# Stochastic Optimization: Theory and Applications

Guest Editors: Bin Wang, Lotfollah Najjar, Neal N. Xiong, and Rung Ching Chen



**Stochastic Optimization: Theory and Applications** 

# **Stochastic Optimization: Theory and Applications**

Guest Editors: Bin Wang, Lotfollah Najjar, Neal N. Xiong, and Rung Ching Chen

Copyright @ 2013 Hindawi Publishing Corporation. All rights reserved.

This is a special issue published in "Journal of Applied Mathematics." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# **Editorial Board**

Saeid Abbasbandy, Iran Mina B. Abd-El-Malek, Egypt Mohamed A. Abdou, Egypt Subhas Abel, India Mostafa Adimy, France Carlos J. S. Alves, Portugal Mohamad Alwash, USA Igor Andrianov, Germany Sabri Arik, Turkey Francis T.K. Au, Hong Kong Olivier Bahn, Canada Roberto Barrio, Spain Alfredo Bellen, Italy Jafar Biazar, Iran Hester Bijl, The Netherlands Anjan Biswas, Saudi Arabia Stephane P.A. Bordas, USA James Robert Buchanan, USA Alberto Cabada, Spain Xiao Chuan Cai, USA **Jinde Cao**, China Alexandre Carvalho, Brazil Song Cen, China Qianshun S. Chang, China Tai-Ping Chang, Taiwan Shih-sen Chang, China Rushan Chen, China Xinfu Chen, USA Ke Chen, UK Eric Cheng, Hong Kong Francisco Chiclana, UK Jen-Tzung Chien, Taiwan C. S. Chien, Taiwan Han H. Choi, Republic of Korea Tin-Tai Chow, China M. S. H. Chowdhury, Malaysia Carlos Conca, Chile Vitor Costa, Portugal Livija Cveticanin, Serbia Eric de Sturler, USA Orazio Descalzi, Chile Kai Diethelm, Germany Vit Dolejsi, Czech Republic Bo-Qing Dong, China Magdy A. Ezzat, Egypt

Meng Fan, China Ya Ping Fang, China Antonio J. M. Ferreira, Portugal Michel Fliess, France M. A. Fontelos, Spain Huijun Gao, China Bernard J. Geurts, The Netherlands Jamshid Ghaboussi, USA Pablo González-Vera, Spain Laurent Gosse, Italy K. S. Govinder, South Africa Jose L. Gracia, Spain Yuantong Gu, Australia Zhihong GUAN, China Nicola Guglielmi, Italy Frederico G. Guimarães, Brazil Vijay Gupta, India Bo Han, China Maoan Han, China Pierre Hansen, Canada Ferenc Hartung, Hungary Xiaoqiao He, Hong Kong Luis Javier Herrera, Spain J. Hoenderkamp, The Netherlands Ying Hu, France Ning Hu, Japan Zhilong L. Huang, China Kazufumi Ito, USA Takeshi Iwamoto, Japan George Jaiani, Georgia Zhongxiao Jia, China Tarun Kant, India Ido Kanter, Israel Abdul Hamid Kara, South Africa Hamid Reza Karimi, Norway Jae-Wook Kim, UK Jong Hae Kim, Republic of Korea Kazutake Komori, Japan Fanrong Kong, USA Vadim . Krysko, Russia Jin L. Kuang, Singapore Miroslaw Lachowicz, Poland Hak-Keung Lam, UK Tak-Wah Lam, Hong Kong PGL Leach, Cyprus

Yongkun Li, China Wan-Tong Li, China J. Liang, China Ching-Jong Liao, Taiwan Chong Lin, China Mingzhu Liu, China Chein-Shan Liu, Taiwan Kang Liu, USA Yansheng Liu, China Fawang Liu, Australia Shutian Liu, China Zhijun Liu, China Julián López-Gómez, Spain Shiping Lu, China Gert Lube, Germany Nazim Idrisoglu Mahmudov, Turkev Oluwole Daniel Makinde, South Africa Francisco J. Marcellán, Spain Guiomar Martín-Herrán, Spain Nicola Mastronardi, Italy Michael McAleer, The Netherlands Stephane Metens, France Michael Meylan, Australia Alain Miranville, France Ram N. Mohapatra, USA Iaime E. Munoz Rivera, Brazil Javier Murillo, Spain Roberto Natalini, Italy Srinivasan Natesan, India Jiri Nedoma, Czech Republic Jianlei Niu, Hong Kong Roger Ohayon, France Javier Oliver, Spain Donal O'Regan, Ireland Martin Ostoja-Starzewski, USA Turgut Öziş, Turkey Claudio Padra, Argentina Reinaldo Martinez Palhares, Brazil Francesco Pellicano, Italy Juan Manuel Peña, Spain Ricardo Perera, Spain Malgorzata Peszynska, USA James F. Peters, Canada Mark A. Petersen, South Africa Miodrag Petkovic, Serbia

Vu Ngoc Phat, Vietnam Andrew Pickering, Spain Hector Pomares, Spain Maurizio Porfiri, USA Mario Primicerio, Italy Morteza Rafei, The Netherlands Roberto Renò, Italy Jacek Rokicki, Poland Dirk Roose, Belgium Carla Roque, Portugal Debasish Roy, India Samir H. Saker, Egypt Marcelo A. Savi, Brazil Wolfgang Schmidt, Germany Eckart Schnack, Germany Mehmet Sezer, Turkey Naseer Shahzad, Saudi Arabia Fatemeh Shakeri, Iran Jian Hua Shen, China Hui-Shen Shen, China Fernando Simões, Portugal Theodore E. Simos, Greece

Abdel-Maksoud A. Soliman, Egypt Qing-Wen Wang, China Xinyu Song, China Qiankun Song, China Yuri N. Sotskov, Belarus Peter J. C. Spreij, The Netherlands Niclas Strömberg, Sweden Ray KL Su, Hong Kong Jitao Sun, China Wenyu Sun, China XianHua Tang, China Alexander Timokha, Norway Mariano Torrisi, Italy Jung-Fa Tsai, Taiwan Ch Tsitouras, Greece Kuppalapalle Vajravelu, USA Alvaro Valencia, Chile Erik Van Vleck, USA Ezio Venturino, Italy Jesus Vigo-Aguiar, Spain Michael N. Vrahatis, Greece Baolin Wang, China Mingxin Wang, China

Guangchen Wang, China Junjie Wei, China Li Weili, China Martin Weiser, Germany Frank Werner, Germany Shanhe Wu, China Dongmei Xiao, China Gongnan Xie, China Yuesheng Xu, USA Suh-Yuh Yang, Taiwan Bo Yu, China Jinyun Yuan, Brazil Alejandro Zarzo, Spain Guisheng Zhai, Japan Jianming Zhan, China Zhihua Zhang, China Jingxin Zhang, Australia Shan Zhao, USA Chongbin Zhao, Australia Renat Zhdanov, USA Hongping Zhu, China

# Contents

**Stochastic Optimization: Theory and Applications**, Bin Wang, Lotfollah Najjar, Neal N. Xiong, and Rung Ching Chen Volume 2013, Article ID 949131, 2 pages

**Study on Indefinite Stochastic Linear Quadratic Optimal Control with Inequality Constraint**, Guiling Li and Weihai Zhang Volume 2013, Article ID 805829, 9 pages

**On Iterative Learning Control for Remote Control Systems with Packet Losses**, Chunping Liu, Rong Xiong, Jianxin Xu, and Jun Wu Volume 2013, Article ID 245372, 9 pages

**Pareto Optimal Solutions for Stochastic Dynamic Programming Problems via Monte Carlo Simulation**, R. T. N. Cardoso, R. H. C. Takahashi, and F. R. B. Cruz Volume 2013, Article ID 801734, 9 pages

**Dynamic Mean-Variance Model with Borrowing Constraint under the Constant Elasticity of Variance Process**, Hao Chang and Xi-min Rong Volume 2013, Article ID 348059, 8 pages

A Fast Optimization Method for Reliability and Performance of Cloud Services Composition Application, Zhao Wu, Naixue Xiong, Yannong Huang, Qiong Gu, Chunyang Hu, Zhongbo Wu, and Bo Hang Volume 2013, Article ID 407267, 12 pages

**Reflected Backward Stochastic Differential Equations Driven by Countable Brownian Motions**, Pengju Duan, Min Ren, and Shilong Fei Volume 2013, Article ID 729636, 7 pages

**An Improved Hybrid Genetic Algorithm with a New Local Search Procedure**, Wen Wan and Jeffrey B. Birch Volume 2013, Article ID 103591, 10 pages

Layer-Based Data Aggregation and Performance Analysis in Wireless Sensor Networks, Hongju Cheng, Yongzhao Chen, Naixue Xiong, and Feifei Li Volume 2013, Article ID 502381, 12 pages

**Doubly Constrained Robust Blind Beamforming Algorithm**, Xin Song, Jingguo Ren, and Qiuming Li Volume 2013, Article ID 245609, 8 pages

Adaptive Waveform Design for Multiple Radar Tasks Based on Constant Modulus Constraint, Wang Bin, Yang Wenfang, and Wang Jinkuan Volume 2013, Article ID 295493, 6 pages **Optimal Waveform Selection for Robust Target Tracking**, Fengming Xin, Jinkuan Wang, Qiang Zhao, and Yuhuan Wang Volume 2013, Article ID 725058, 7 pages

**Estimating Time-Varying Beta of Price Limits and Its Applications in China Stock Market**, Rongquan Bai, Zuoquan Zhang, and Menggang Li Volume 2013, Article ID 682159, 8 pages

**Qualitative and Quantitative Integrated Modeling for Stochastic Simulation and Optimization**, Xuefeng Yan, Yong Zhou, Yan Wen, and Xudong Chai Volume 2013, Article ID 831273, 12 pages

**Univex Interval-Valued Mapping with Differentiability and Its Application in Nonlinear Programming**, Lifeng Li, Sanyang Liu, and Jianke Zhang Volume 2013, Article ID 383692, 8 pages

**Smoothing Techniques and Augmented Lagrangian Method for Recourse Problem of Two-Stage Stochastic Linear Programming**, Saeed Ketabchi and Malihe Behboodi-Kahoo Volume 2013, Article ID 735916, 8 pages

New Results on Robust Stability and Stabilization of Linear Discrete-Time Stochastic Systems with Convex Polytopic Uncertainties, P. Niamsup and G. Rajchakit Volume 2013, Article ID 368259, 10 pages

# *Editorial* **Stochastic Optimization: Theory and Applications**

# Bin Wang,<sup>1</sup> Lotfollah Najjar,<sup>2</sup> Neal N. Xiong,<sup>3</sup> and Rung Ching Chen<sup>4</sup>

<sup>1</sup> Engineering Optimization and Smart Antenna Institute, School of Computer and Communication Engineering,

Northeastern University at Qinhuangdao, Hebei, Qinhuangdao 066004, China

<sup>2</sup> Information System Quantitative Analysis, College of Information Science & Technology (IS&T),

University of Nebraska at Omaha, Omaha, NE 68182, USA

<sup>3</sup> School of Computer Science, Colorado Technical University, 4435 North Chestnut Street, Colorado Springs, CO 80907, USA

<sup>4</sup> College of Informatics Chaoyang University of Technology, 168 Jifong East Road, Wufong District, Taichung City 41349, Taiwan

Correspondence should be addressed to Bin Wang; wangbinneu@qq.com

Received 23 October 2013; Accepted 23 October 2013

Copyright © 2013 Bin Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an important branch of applied mathematics, optimization theory, especially stochastic optimization, becomes an important tool for solving multiobjective decision-making problems in random process recently. Many kinds of industrial, biological, engineering, and economic problems can be viewed as stochastic systems, for example, area of communication, gene, signal processing, geography, civil engineering, aerospace, banking, and so forth. Stochastic optimization is suitable to solve the decision-making problems in these stochastic systems.

This special issue includes 16 high-quality peer-reviewed papers that deal with different aspects of stochastic optimization problems. These papers contain some new, novel, and innovative techniques and ideas. We hope that all the papers published in this special issue can stimulate the continuing efforts to understand this field, particularly new stochastic optimization algorithms and new applications in related fields.

In the paper entitled "Qualitative and quantitative integrated modeling for stochastic simulation and optimization," the authors propose a qualitative and quantitative combined modeling specification based on a hierarchical model structure framework. The new modeling approach is based on a hierarchical model structure which includes the meta-meta model, the metamodel, and the high-level model.

In the paper entitled "*Estimating time-varying beta of price limits and its applications in China stock market*," the authors propose an estimation method of time-varying beta of price limits. It uses China stock market trading data to estimate time-varying beta and researches on systemic risk in China stock market.

In the paper entitled "*Doubly constrained robust blind beamforming algorithm*," the authors propose doubly constrained robust least squares constant modulus algorithm (LSCMA) to solve the problem of signal steering vector mismatches via the Bayesian method and worst-case performance optimization, which is based on the mismatches between the actual and presumed steering vectors. A theoretical analysis for the proposed algorithm in terms of complexity cost, convergence performance, and SINR performance is presented in this paper.

In the paper entitled "Smoothing techniques and augmented Lagrangian method for recourse problem of twostage stochastic linear programming," the authors apply the smoothing techniques and a fast Newton-Armijo algorithm for solving an unconstrained smooth reformulation of this problem. Computational results and comparisons are given to show the effectiveness and speed of the algorithm.

In the paper entitled "New results on robust stability and stabilization of linear discrete-time stochastic systems with convex polytopic uncertainties," the authors propose new delaydependent mean square robust stability conditions for linear polytopic delay-difference stochastic equations with interval time-varying delays in terms of LMIs. An application to robust stabilization of linear discrete-time stochastic control systems is given in this paper. In the paper entitled "Layer-based data aggregation and performance analysis in wireless sensor networks," the authors focus on the minimum-latency data aggregation problem and propose a new efficient scheme for it. The basic idea is to build an aggregation tree by ordering nodes into layers and then use a scheduling algorithm on the basis of the aggregation tree to determine the transmission time slots for all nodes in the network with collision avoiding.

In the paper entitled "Optimal waveform selection for robust target tracking," the authors assume a one-dimension target model which will try to escape the radar detection to degrade the tracking performance and propose a new optimal waveform selection algorithm based on game theory for robust tracking. The optimal parameters for transmitted waveform are finally found by the minimization of the trace of the estimated state error covariance.

In the paper entitled "Adaptive waveform design for multiple radar tasks based on constant modulus constraint," the authors propose a waveform design method which can efficiently synthesize waveforms and provide a trade-off between estimation performance and detection performance and then apply the technique of generating a constant modulus signal with the given Fourier transform magnitude to the waveform. Finally a waveform that has constant modulus property can be obtained.

In the paper entitled "Univex interval-valued mapping with differentiability and its application in nonlinear programming," the authors introduce interval-valued univex functions for differentiable programming problems and derive optimality and duality results for a class of generalized convex optimization problems with interval-valued univex functions.

In the paper entitled "Study on indefinite stochastic linear quadratic optimal control with inequality constraint," the authors study the indefinite stochastic LQ optimal control problem with unequally terminal state constraint, which can be transformed into a hybrid constrained mathematical programming problem and design a dynamic programming algorithm to solve the constrained indefinite stochastic LQ issue.

In the paper entitled "A fast optimization method for reliability and performance of cloud services composition application," the authors propose a fast optimization method for reliability and performance of cloud services composition application based on Universal Generating Function and Genetic Algorithm. The model and algorithm can be applied in online prediction and optimization for reliability and performance of cloud services composition application.

In the paper entitled "Dynamic mean-variance model with borrowing constraint under the constant elasticity of variance process," the authors study a continuous-time dynamic meanvariance portfolio selection problem with the constraint of a higher borrowing rate, in which stock price is governed by a constant elasticity of variance (CEV) process.

In the paper entitled "An improved hybrid genetic algorithm with a new local search procedure," the authors propose a novel, simplified, and efficient HGA with a new individual learning procedure that performs a LS only when the best offspring (solution) in the offspring population is also the best in the current parent population and develop a new LS method based on a three-directional search (TD), which is derivative-free and self-adaptive.

In the paper entitled "*Reflected backward stochastic differential equations driven by countable brownian motions*," the authors deal with a new class of reflected backward stochastic differential equation driven by countable Brownian motions. The existence and uniqueness of the RBSDEs is obtained via Snell envelope and fixed theorem.

In the paper entitled "Pareto optimal solutions for stochastic dynamic programming problems via Monte Carlo simulation," the authors propose a heuristic algorithm for a class of stochastic discrete-time continuous-variable dynamic programming problems submitted to non-Gaussian disturbances. This new idea is carried out by using Monte Carlo simulations embedded in an approximate algorithm proposed for deterministic dynamic programming problems. The new method is tested in instances of the classical inventory control problem.

In the paper entitled "On iterative learning control for remote control systems with packet loss," the authors propose an ILC for a time-varying system with random packet dropouts. The ILC law adopts an iteration-average operator and a revised learning gain that takes into consideration the probabilities of data-dropout factors.

#### Acknowledgments

As guest editors for this special issue, we would like to take this opportunity to thank all the contributions from the authors and reviewers. Bin Wang was supported by the Natural Science Foundation of Hebei Province (no. F2013501075), the Doctoral Scientific Research Foundation of Liaoning Province (no. 20131030), and the Fundamental Research Funds for the Central Universities (no. N110323005).

> Bin Wang Lotfollah Najjar Neal N. Xiong Rung Ching Chen

# **Research** Article

# **Study on Indefinite Stochastic Linear Quadratic Optimal Control with Inequality Constraint**

# Guiling Li<sup>1</sup> and Weihai Zhang<sup>2</sup>

<sup>1</sup> College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266591, China
 <sup>2</sup> College of Information and Electrical Engineering, Shandong University of Science and Technology, Qingdao 266591, China

Correspondence should be addressed to Weihai Zhang; w\_hzhang@163.com

Received 1 April 2013; Revised 12 September 2013; Accepted 2 October 2013

Academic Editor: Lotfollah Najjar

Copyright © 2013 G. Li and W. Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies the indefinite stochastic linear quadratic (LQ) optimal control problem with an inequality constraint for the terminal state. Firstly, we prove a generalized Karush-Kuhn-Tucker (KKT) theorem under hybrid constraints. Secondly, a new type of generalized Riccati equations is obtained, based on which a necessary condition (it is also a sufficient condition under stronger assumptions) for the existence of an optimal linear state feedback control is given by means of KKT theorem. Finally, we design a dynamic programming algorithm to solve the constrained indefinite stochastic LQ issue.

# 1. Introduction

The study on LQ control problems can be traced back to the pioneering work of Kalman [1] and Wonham [2] several decades ago. The LQ control theory is elegantly established and developed, and the main work can be seen in [3–11]. In particular, it is found [6] that a stochastic LQ problem with indefinite control weighting matrices may still be well-posed. This discovery evokes a series of subsequent researches, and many important achievements are obtained [5, 12–16]. Up to now, most work deals with the indefinite stochastic LQ problems without constraints. However, as a practical optimization problem, the indefinite stochastic LQ problem unavoidably has various constraints on the state or control; in particulary the inequality constraints often appear.

For the constrained indefinite stochastic LQ problems, [17] studied the equally constrained stochastic LQ optimization for Itô systems. In this paper, we will study the stochastic LQ problem with inequality constraint.

Firstly, we present and prove the generalized KKT theorem under hybrid constraints. Secondly, a necessary condition for the existence of an optimal linear state feedback control is given by means of the generalized KKT theorem. Thirdly, if we strengthen the condition, we can obtain a necessary and sufficient condition for the existence of the optimal linear feedback control to indefinite stochastic LQ optimal control problem with inequality constraint. Finally, we give a dynamic programming algorithm to solve the stochastic LQ problem with the inequality constraint. We provide an example to demonstrate the effectiveness of our main theoretical results.

The outline of this paper is organized as follows. In Section 2, we present a generalized KKT theorem under hybrid constraints. Section 3 proposes a KKT condition for the existence of an optimal linear state feedback control. In Section 4, we provide a necessary and sufficient condition and a dynamic programming algorithm for the stochastic LQ problem with inequality constraint. Section 5 concludes the paper.

For convenience, throughout the paper, we adopt the following notations:  $A^T$  denotes the transpose of a matrix A. A > 0 ( $A \ge 0$ ): A is a positive definite (positive semidefinite) symmetric matrix. tr(A): trace of a square matrix.  $R^{m \times n}$ : the space of all  $m \times n$  real matrices.  $S^n$ : a  $n \times n$  symmetric matrix space.

#### 2. Preliminaries

Consider the following indefinite stochastic LQ control.

Problem 1. Consider

min 
$$J(x_0, u) = E \int_0^T \left[ x^T(t) Q(t) x(t) + u^T(t) R(t) u(t) \right] dt,$$
 (1a)

s.t. 
$$dx(t) = [A(t)x(t) + B(t)u(t)]dt$$
  
+  $[C(t)x(t) + D(t)u(t)]dw(t).$  (1b)

$$x\left(0\right) = x_0,\tag{1c}$$

$$E\left\{ \|x(T)\|^{2} \right\} = E\left[x^{T}(T)x(T)\right] \le c,$$
 (1d)

where  $x(t)=[x_1(t), x_2(t), ..., x_n(t)]^T$  is an *n*-dimensional state variable,  $u(t) \in \mathbb{R}^m$  is a control input,  $w(\cdot)$  is a onedimensional standard Brownian motion defined on a filtered probability space  $(\Omega, F, F_t, P)$ . We denote the information flow  $\mathscr{F}_t = \sigma[w(s): 0 \le s \le t \le T]$ .  $u(\cdot)$  belongs to  $\mathscr{L}^2_{\mathscr{F}}(\mathbb{R}^m)$ , where  $\mathscr{L}^2_{\mathscr{F}}(\mathbb{R}^m)$  is a space of all  $\mathbb{R}^m$ -valued,  $\mathscr{F}_t$ -adapted measurable processes satisfying  $E \int_0^T \|u(t)\|^2 dt < +\infty$ . For each admissible control, the corresponding trajectory satisfies the constraint (1c). c in constraint (1c) is a given nonnegative constant. A(t), B(t), C(t), D(t), Q(t), and R(t) are timevarying matrices of suitable dimensions. Q(t) and R(t) in objective functional are symmetric matrices. To study the issue, we first put forward the following Assumption H<sub>1</sub>.

Assumption  $H_1$ .  $A(t), C(t) \in \mathscr{L}^{\infty}(0, T, \mathbb{R}^{n \times n}), Q(t) \in \mathscr{L}^{\infty}(0, T, S^n), B(t), D(t) \in \mathscr{L}^{\infty}(0, T, \mathbb{R}^{n \times m}), \text{ and } \mathbb{R}(t) \in \mathscr{L}^{\infty}(0, T, S^m), \text{ where } L^{\infty}(0, T, X) := \{f(t) : X \text{-valued essential bounded measurable function and ess sup}_{t \in [0,T]} ||f(t)|| < +\infty \}.$ 

In this paper, the weighting matrices in the objective functional are not required to be definite. Therefore Problem 1 is an indefinite stochastic LQ optimal control problem. For later use, we recall KKT theorem for this type of mathematical programming (MP) problems:

min 
$$f(x)$$
  
s.t.  $g(x) \le 0$ , (2)  
 $h(x) = 0$ ,

where  $\mathbf{g}(x) = (g_1(x), \dots, g_p(x))$ , and  $\mathbf{h}(x) = (h_1(x), \dots, h_q(x))$ .

The KKT conditions [18–20], which are also known as the Kuhn-Tucker (KT) conditions, are the first-order necessary conditions for a solution in nonlinear programming to be optimal, provided that some regularity conditions are satisfied. The Lagrange multiplier method, which allows only equality constraints, can be viewed as a special case of KKT conditions.

*Regularity Condition (or Constraint Qualification).* In MP above, let  $I^* = \{i \mid g_i(x^*) = 0\}$   $(g_i(x^*) i \in I^*$  be active constraints at  $x^*$ ). The gradient vectors  $\nabla g_i(x^*)$ ,  $i \in I^*$ , and

 $\nabla h_j(x^*)$ , j = 1, ..., q, are linearly independent, is known as a linear independent constraint qualification (LICQ).

*Regular Point.* In MP above,  $x^*$  is said to be a regular point of the constraints if the gradient vectors  $\nabla g_i(x^*)$ ,  $\nabla h_j(x^*)$ ,  $i \in I^*$ , j = 1, ..., q, are linearly independent.

*KKT Theorem.* In MP above, we assume that the functions f,  $\mathbf{g} = (g_1, \ldots, g_p)^T$ ,  $\mathbf{h} = (h_1, \ldots, h_q)^T$  are twice continuously differentiable and we assume that all the constraints satisfy the regularity condition LICQ. Let  $x^*$  be a point satisfying all the constraints and let  $x^*$  be a regular point of the above constraints. Now suppose that this regular point  $x^*$  is also a relative minimum point for the original MP. Then it is shown that there exist a vector  $\lambda \ge 0 \in R$  and a vector  $\mu \in R$ , such that

$$\nabla_{x}L\left(x^{*},\lambda^{*},\mu^{*}\right) = \mathbf{0},$$

$$\lambda_{i}^{*}g_{i}\left(x^{*}\right) = 0, \quad i = 1,\dots, p,$$
(3)

where  $L(x, \lambda, \mu) = f(x) + \lambda^T \mathbf{g}(x) + \mu^T \mathbf{h}(x)$  is the Lagrangian function and  $\lambda_i^* g_i(x^*) = 0$ , i = 1, ..., p, are complementary clackness condition.

It is particularly important to check the regularity condition before we apply the conclusion of KKT theorem. If it is not so, the conclusion of KKT theorem would not be valid, just as the following example shows.

Example 2. Consider

min 
$$f(x_1, x_2) = -2x_1 - 3x_2,$$
  
s.t.  $\mathbf{g}(x_1, x_2) = x_1^2 + x_2^2 = 0.$  (4)

Obviously, the minimum point is  $x^*(0,0)^T$ . According to KKT theorem, we obtain

$$2 - \lambda x_1^* = 0,$$
  

$$3 - \lambda x_2^* = 0,$$
 (5)  

$$x_1^{*2} + x_2^{*2} = 0.$$

The conclusion of KKT theorem does not hold at point  $x^*(0,0)^T$ , because  $\nabla \mathbf{g}(0,0) = (0,0)^T$  is not linearly independent. It does not satisfy the LICQ regularity condition.

In order for a minimum point  $x^*$  to satisfy the above KKT conditions, it should satisfy some regularity conditions. Except for LICQ regularity condition, the most used ones are listed below.

Constant Rank Constraint Qualification. For each subset of the gradients of the active inequality constraints and the gradients of the equality constraints the rank at a vicinity of  $x^*$  is constant.

*Mangasarian-Fromovitz Constraint Qualification.* The gradients of the active inequality constraints and the gradients of the equality constraints are linear independent at  $x^*$ .

*Constant Positive Linear Dependence Constraint Qualification.* For each subset of the gradients of the active inequality constraints and the gradients of the equality constraints, if it is positive-linear dependent at  $x^*$ , then it is positive-linear dependent at a vicinity of  $x^*$ .

The Slater condition for a convex MP is also a common regularity condition.

*Remark 3.* In this paper, for convenience, when we use the KKT theorem, we always assume that the local optimal  $x^*$  meets the LICQ regularity condition. The same goes for other regularity conditions.

*Definition 4* (see [21]). Let X be a vector space, Y a normed space, and T a transformation from X to Y. If the limit

$$\delta T(x;h) = \lim_{\alpha \to 0} \frac{\left[T(x+\alpha h) - T(x)\right]}{\alpha}$$
(6)

exists, it is called the Gateaux differential of *T* at *x* with increment *h*. If the limit exists for each  $h \in X$ , the transformation *T* is said to be Gateaux differentiable at *x*.

Definition 5 (see [21]). Let X be a vector space and Z a Banach space with a positive cone P having nonempty interior. Let G be a mapping from X to Z which has a Gateaux differential that is linear in its increment. A point  $x_0 \in X$  is said to be a regular point of the inequality  $G(x) \le 0$ , if  $G(x_0) \le 0$  and there is an  $h \in X$  such that  $G(x_0) + \delta G(x_0; h) < 0$ .

Definition 6 (see [21]). Let X be a vector space and Z a Banach space. Let  $H(x) = [h_1(x), \ldots, h_1(x)]^T$  be a mapping from X to Z which has a Gateaux differential that is linear in its increment. A point  $x_0 \in X$  is said to be a regular point of the equality H(x) = 0, if  $\delta h_1(x_0; h), \ldots, \delta h_n(x_0; h)$  are linearly independent.

On the basis of the definitions above, let us discuss the KKT theorem in Banach space, where the objective function and the constraint functions in MP are functionals.

Let us consider

min 
$$f(x)$$
  
s.t.  $G(x) \le 0$  (MP I)  
 $H(x) = 0.$ 

As a special case,

min 
$$f(x)$$
  
s.t.  $G(x) \le 0$  (MP II)

has the local necessary condition as follows.

**Lemma 7** (see [21] (generalized KKT theorem)). Let X be a vector space and Z a Banach space having positive cone P. Assume that P contains an interior point. Let f be a Gateaux differentiable functional on X and G a Gateaux differentiable mapping from X to Z. Assume that the Gateaux differentials are linear in their increments. Suppose that  $x_0$  minimizes f subject to  $G(x) \leq 0$  and that  $x_0$  is a regular point of the inequality  $G(x) \leq 0$ . Then there is a  $z_0^* \geq 0$  in Z, such that *The following theorem is the local necessary condition of (MP I).* 

**Theorem 8.** Let X be a vector space and Z a Banach space having positive cone P. Assume that P contains an interior point. Let f be a Gateaux differentiable functional on X. Let G and H be Gateaux differentiable mappings from X to Z. Assume that the Gateaux differentials are linear in their increments. Suppose that  $x_0$  minimizes f subject to  $G(x) \le 0$ , H(x) = 0 and that  $x_0$  is a regular point of  $G(x) \le 0$ , H(x) = 0. Then there is a  $\lambda^* \ge 0$  in Z,  $\mu^* \in Z$ , such that the Lagrangian function  $f(x) + \lambda^* G(x) + \mu^* H(x)$  is stationary at  $x_0$ . Namely,  $\delta f(x_0; h) + \lambda^* \delta G(x_0; h) + \mu^* \delta H(x_0; h) = 0$ . Furthermore,  $z_0^* G(x_0) = 0$ .

*Proof.* H(x) = 0 is equivalent to  $H(x) \le 0$  and  $-H(x) \le 0$ . If  $x_0$  is a regular point of H(x) = 0, then  $\delta h_1(x_0; h), \ldots, \delta h_n(x_0; h)$  are linearly independent. So  $\delta h_1(x_0; h), \ldots, \delta h_n(x_0; h)$  are all nonzero, because the Gateaux differentials are linear in their increments. Using Definition 5, it is easy to verify that  $x_0$  is a regular point of both  $-H(x) \le 0$  and  $H(x) \le 0$ . According to Lemma 7, we know that the multiplier  $\mu^*$  of equality H(x) = 0 has no nonnegative requirement.

*Definition 9* (see [22]). Suppose that f(X) is a scalar-valued function of the elements  $x_{ij}$  of X. Then the gradient matrix of f(X) is defined as

$$\nabla \left[ f\left( X\right) \right] = \frac{\partial f\left( X\right) }{\partial \left( X\right) } \tag{7}$$

with

$$\left[\frac{\partial f(X)}{\partial (X)}\right]_{ij} = \frac{\partial f(X)}{\partial x_{ij}}.$$
(8)

Based on Definition 9, we can easily extend KKT theorem from Banach space to matrix space. Because X(t) can be treated as a vector  $[X_1(t), \ldots, X_n(t)]^T$ , one can work out  $\nabla[X_i(t)]$  and the KKT theorem holds.

When we apply the matrix KKT theorem, we need to give the partial list of gradient matrices [22] that we will use in this paper. In the following formulas, X is an  $n \times m$  matrix. The formulas are not valid if the elements  $x_{ij}$  of X are not independent. A, B are assumed to have appropriate dimensions determined from context.

Consider the following:

2

$$\frac{\partial}{\partial X} \operatorname{tr} (X) = I,$$

$$\frac{\partial}{\partial X} \operatorname{tr} (XX^{T}) = 2X,$$

$$\frac{\partial}{\partial X} \operatorname{tr} (AX^{T}) = A,$$

$$\frac{\partial}{\partial X} \operatorname{tr} (AX) = A^{T},$$

$$\frac{\partial}{\partial X} \operatorname{tr} (AX^{T}B) = BA,$$

$$\frac{\partial}{\partial X} \operatorname{tr} (AXB) = A^{T}B^{T},$$

$$\frac{\partial}{\partial X} \operatorname{tr} (AXBX^{T}) = A^{T}XB^{T} + AXB,$$

$$\frac{\partial}{\partial X} \operatorname{tr} (AXBX) = A^{T}X^{T}B^{T} + B^{T}X^{T}A^{T}.$$
(9)

# 3. KKT Conditions and a New Type of GDREs

*Definition 10.* Problem 1 is well-posed, if for any  $x_0 \in \mathbb{R}^n$ ,  $-\infty < V(x_0) = \inf_{u(\cdot) \in U_{ad}} \{J(x_0, u(\cdot))\}$ .  $u_*$  is called an optimal control, if  $V(x_0) = J(x_0, u_*)$ , and  $x_*$  denotes the corresponding optimal trajectory.

Let the control law be u = Kx ( $K \in C^{m \times n}[0,T]$ ), and  $X(t) = E[x(t)x^{T}(t)]$ . By substituting u = Kx into (1a) of Problem 1, we obtain the new objective functional:

$$J(X,K) = \operatorname{tr} \int_0^T \left[ QX + K^T R K X \right] dt, \qquad (10)$$

where  $C^{n \times n}[0, T]$  is the space of *n*-order square matrix whose elements are continuous functions. By substituting u = Kx into (1b) of Problem 1, we obtain a closed-loop system:

$$dx = (A + BK) x dt + (C + DK) x dw, \quad x (0) = x_0.$$
 (11)

By applying Itô's formula to X(t), we obtain

$$\dot{X} = (A + BK) X + X(A + BK)^{T} + (C + DK) X(C + DK)^{T},$$
(12)  
$$X (0) = X_{0} = x_{0} x_{0}^{T}.$$

Define the transformation H(X, K) from  $C^{n \times n}[0, T] \times C^{m \times n}[0, T]$  to  $C^{n \times n}[0, T]$ :

$$H(X,K) := X(t) - X(0) - \int_{0}^{t} \left[ (A + BK) X + X (A + BK)^{T} + (C + DK) X (C + DK)^{T} \right] dt.$$
(13)

By substituting u = Kx into (1c) of Problem 1, we obtain

$$\operatorname{tr} X\left(T\right) - c \le 0. \tag{14}$$

Define the transformation G(X, K) from  $C^{n \times n}[0, T]$  to  $R^1$ :

$$G(X(T)) := \operatorname{tr} X(T) - c.$$
 (15)

So the original stochastic Problem 1 can be transformed into the deterministic Problem 11 as follows. Problem 11. Consider the following:

min 
$$J(X,K) = \operatorname{tr} \int_0^T \left[ QX + K^T R K X \right] dt,$$
 (16a)

s.t. 
$$H(X, K) = 0, \quad \forall t \in [0, T],$$
 (16b)

$$G\left(X\left(T\right)\right) \le 0. \tag{16c}$$

**Lemma 12.** J(X, K), H(X, K), and G(X(T)) have continuous *Gateaux derivative as follows:* 

$$\begin{split} \delta J_X\left(X,K;\Delta X\right) &= \mathrm{tr} \int_0^T \left(Q + K^T R K\right) \Delta X dt, \\ \delta J_K\left(X,K;\Delta X\right) &= \mathrm{tr} \int_0^T \left(\Delta K^T R K X + K^T R \Delta K X\right) dt, \\ \delta H_X\left(X,K;\Delta X\right)(t) \end{split}$$

$$= \Delta X (t) - \operatorname{tr} \int_0^t \left[ (A + BK) \Delta X + \Delta X (A + BK)^T + (C + DK) \Delta X (C + DK)^T \right] dt,$$

$$\delta H_{K}(X, K; \Delta K)(t)$$

$$= -\operatorname{tr} \int_{0}^{t} \left[ B\Delta KX + X\Delta K^{T}B^{T} + (C + DK) X(D\Delta K)^{T} + (D\Delta K) X(C + DK)^{T} \right] dt,$$

$$\delta G_{X}(X; \Delta X(T)) = \operatorname{tr} \Delta X(T).$$
(17)

*Proof.* We prove only the most complicated one. The rest can be verified in the same way. From Definition 4,

$$\begin{split} \delta H_X(X,K;\Delta X) &= \lim_{\alpha \to 0} \frac{\left[H\left(X + \alpha \Delta X, K\right) - H\left(X, K\right)\right]}{\alpha}, \\ H\left(X,K\right) \\ &:= X\left(T\right) - X\left(0\right) - \int_0^t \left[\left(A + BK\right)X + X(A + BK\right)^T + \left(C + DK\right)X(C + DK)^T\right] dt. \end{split}$$

Replace the *X* in *H*(*X*, *K*) with  $(X + \alpha \Delta X)$  and then let  $\alpha \rightarrow 0$  showing the conclusion.

(18)

**Lemma 13** (see [21]). If  $\alpha(t)$  and  $\beta(t)$  are continuous in  $[t_1, t_2]$ and  $\int_{t_1}^{t_2} [\alpha(t)h(t) + \beta(t)h(t)]dt = 0$  for every continuously differentiable h(t) with  $h(t_1) = h(t_1) = 0$ , then  $\beta$  is differentiable and  $\alpha(t) \equiv \beta(t)$  in  $[t_1, t_2]$ . **Lemma 14** (see [21]). If  $\alpha(t)$  is continuous in  $[t_1, t_2]$  and  $\int_{t_1}^{t_2} [\alpha(t)h(t)]dt = 0$  for every continuously differentiable h(t) with  $h(t_1) = h(t_1) = 0$ , then  $\alpha(t) \equiv 0$  on  $[t_1, t_2]$ .

**Theorem 15.** Assume that  $K_*$  is the optimal solution of Problem 1, and then there exist a symmetric matrix  $P \in NBV^{n \times n}$ [0, T] and a nonnegative  $\lambda \in \mathbb{R}^1$ , such that

$$-\dot{P} = (Q + K_*^T R K_*) + P (A + B K_*) + (A + B K_*)^T P$$
  
+ (C + DK\_\*) P(C + DK\_\*)^T, (19a)

$$P(T) = \lambda I_n, \tag{19b}$$

$$K_*^T R + PB + (C + DK_*)^T PD = 0,$$
 (19c)

$$\lambda \left[ \operatorname{tr} X \left( T \right) - c \right] = 0, \tag{19d}$$

where  $NBV^{n\times n}[0,T]$  is a matrix space whose elements are bounded functions in [0,T] with 0 value at point 0 and right continuous at (0,T).

*Proof.*  $K_*$  is also the optimal solution of Problem 11. Problem 11 is the type of (MP I). Assume that the optimal solution to Problem 11 is  $(X_*, K_*)$ . Using Theorem 8 and Lemma 12, there exist a symmetric matrix  $P \in NBV^{n \times n}[0, T]$  and a nonnegative  $\lambda \in R^1$ , such that

$$\operatorname{tr} \int_{0}^{T} \left( Q + K_{*}^{T} R K_{*} \right) \Delta X dt$$
$$+ \operatorname{tr} \int_{0}^{T} \left\{ \Delta X \left( t \right) \right.$$
$$- \int_{0}^{t} \left[ \left( A + B K_{*} \right) \Delta X + \Delta X \left( A + B K_{*} \right)^{T} \right.$$
$$+ \left( C + D K_{*} \right) \Delta X \left( C + D K_{*} \right)^{T} \right] dt \right\} dF$$

$$+\lambda \operatorname{tr}\left[\Delta X\left(T\right)\right] = 0,\tag{20}$$

$$\operatorname{tr} \int_{0}^{T} \left( \Delta K^{T} R K_{*} X_{*} + K_{*}^{T} R \Delta K K_{*} \right) dt$$
$$- \operatorname{tr} \int_{0}^{T} \left\{ \int_{0}^{t} \left[ B \Delta K X_{*} + X_{*} \Delta K^{T} B^{T} + \left( C + D K_{*} \right) X_{*} \left( D \Delta K \right)^{T} + \left( D \Delta K \right) X_{*} \right.$$
$$\left. \times \left( C + D K_{*} \right)^{T} \right] dt \right\} dP = 0, \tag{21}$$

 $\lambda \left[ \operatorname{tr} X \left( T \right) - c \right] = 0. \tag{22}$ 

For all  $(\Delta X, \Delta K) \in C^{n \times n}[0, T] \times C^{m \times n}[0, T]$ , (20)-(21) are established. According to Riesz representation theorem, we obtain the second item of (20) and the same of (21).

Without loss of generality, let p(T) = 0. Integrate (20) by parts yielding

$$\operatorname{tr} \int_{0}^{T} \left( Q + K_{*}^{T} R K_{*} \right) \Delta X dt + \operatorname{tr} \int_{0}^{T} \Delta X dP$$
$$+ \operatorname{tr} \int_{0}^{T} \left\{ P \left[ \left( A + B K_{*} \right) \Delta X + \Delta X \left( A + B K_{*} \right)^{T} \right. \right.$$
$$\left. + \left( C + D K_{*} \right)^{T} \Delta X \left( C + D K_{*} \right)^{T} \right] \right\} dt$$
$$+ \lambda \operatorname{tr} \left[ \Delta X \left( T \right) \right] = 0.$$
(23)

Clearly, *P* has no jump on [0, T). But *P* has a jump at *T*, and the value is  $-\lambda I_n$ . Because the above results are established for all continuous  $\Delta X$ , then

$$\int_0^T \Delta X dP = P \Delta X \big|_0^T - \int_0^T P \Delta \dot{X} dt = -\int_0^T P \Delta \dot{X} dt.$$
(24)

Thus, (20) becomes

$$\operatorname{tr} \int_{0}^{T} \left[ \left( Q + K_{*}^{T} R K_{*} \right) \Delta X + P \left( A + B K_{*} \right) \Delta X + \left( A + B K_{*} \right)^{T} P \Delta X + \left( C + D K_{*} \right)^{T} P \left( C + D K_{*} \right)^{T} \Delta X - P \Delta \dot{X} \right] dt = 0.$$

$$(25)$$

From Lemma 13, P is differential in [0, T) and (19a) is obtained.

In the same way, integrating (21) by parts, we obtain

$$\operatorname{tr} \int_{0}^{T} \left[ K_{*}^{T} R \left( \Delta K X_{*} \right) + PB \left( \Delta K X_{*} \right) + \left( C + D K_{*} \right)^{T} PD \left( \Delta K X_{*} \right) \right] dt = 0.$$

$$(26)$$

From Lemma 14, (19c) is obtained.

To ensure the continuity of *P*, replace P(T) = 0 with  $P(T) = \lambda I_n$  (i.e., (19b)).

Equation (19d) is called complementary slackness conditions.  $\hfill \Box$ 

*Remark 16.* Equations (16b)-(16c) of Problem 11 and (19a)-(19c) of Theorem 15 are 2*n*-dimensional, first-order differential equations including 2*n* terminal conditions and *m* algebraic equations. Equation (19d) is called a complementary slackness condition. By using these conditions,  $X_*, K_*, P$ , and  $\lambda$  are obtained.

*Remark 17.* As for the complementary slackness condition, if the inequality constraint of Problem 11 is strict, then  $\lambda = 0$ , and the problem becomes easier. If the inequality constraint of Problem 11 is an equality constraint, it simplifies Theorem 15 as Lagrange multiplier method.

*Definition 18.*  $M \in \mathbb{R}^{n \times n}$  is a given matrix. One calls  $M^+$  the Moore-Penrose generalized inverse of M, if

$$MM^{\dagger}M = M, \qquad M^{\dagger}MM^{\dagger} = M^{\dagger},$$
  
$$\left(MM^{\dagger}\right)^{T} = MM^{\dagger}, \qquad \left(M^{\dagger}M\right)^{T} = M^{\dagger}M.$$
  
(27)

Based on Definition 18, we can rewrite Theorem 15 by expressing  $K_*$  in terms of P.

**Lemma 19** (see [13]). *Let matrices L, M, and N be given with appropriate sizes. Then the matrix equation* 

$$LXM = N \tag{28}$$

has a solution X if and only if

$$LL^{\dagger}NM^{\dagger}M = N. \tag{29}$$

Moreover, any solution to LXM = N is represented by

$$X = L^{\dagger} N M^{\dagger} + S - L^{\dagger} L S M M^{\dagger}, \qquad (30)$$

where *S* is a matrix with an appropriate size.

**Theorem 20.** If  $K_*$  is optimal solution of Problem 1, then there exist a unique  $P \in NBV^{n \times n}[0,T]$  and a nonnegative  $\lambda \in R^1$ , such that

$$-\dot{P} = \left(Q + K_*^T R K_*\right) + P \left(A + B K_*\right) \\ + \left(A + B K_*\right)^T P + \left(C + D K_*\right) P \left(C + D K_*\right)^T, \\ P \left(T\right) = \lambda I_n, \\ K_*^T R + P B + \left(C + D K_*\right)^T P D = 0, \quad (31) \\ \lambda \left[ \operatorname{tr} X \left(T\right) - c \right] = 0, \\ K_* = - \left(R + D^T P D\right)^\dagger \left(B^T P + D^T P C\right) \\ + Y - \left(R + D^T P D\right)^\dagger \left(R + D^T P D\right) Y,$$

where  $Y \in L^2(0, T; \mathbb{R}^{m \times n})$ .

Proof. Form (19c) in Theorem 15, we obtain

$$\left(R+D^{T}PD\right)K_{*}=-\left(B^{T}P+D^{T}PC\right).$$
(32)

According to Lemma 19,

$$K_* \left( R + D^T P D \right)^+ \left( B^T P + D T P C \right) + Y$$

$$- \left( R + D^T P D \right)^+ \left( B^T P + D^T P C \right) Y,$$
(33)

where  $Y \in L^2(0, T; \mathbb{R}^{m \times n})$ .

As a special case, let us consider the following discrete stochastic LQ control problem without inequality constraint. Problem 21. Consider the following.

min 
$$J(X, K) = \operatorname{tr} \int_0^T \left[ QX(t) + K(t)^T RK(t) X(t) \right] dt$$
  
s.t.  $\dot{X} = (A + BK) X + X(A + BK)^T$ 

$$+ (C + DK) X + A(A + DK)^{T}$$
$$X (0) = X_{0}.$$
(34)

**Corollary 22.** If  $K_*$  is optimal solution of Problem 21, then there exists a unique  $P \in NBV^{n \times n}[0,T]$  to the following constrained GDRE [13]

$$-\dot{P} = PA + AP^{T} + C^{T}PC + Q$$

$$- \left(PB + C^{T}PD\right) \left(R + D^{T}PD\right) \left(PB + C^{T}PD\right)^{T},$$

$$P (T) = 0,$$

$$\left(R + D^{T}PD\right) \left(R + D^{T}PD\right)^{+} \left(B^{T}P + D^{T}PC\right)$$

$$= B^{T}P + D^{T}PC,$$

$$\left(R + D^{T}PD\right) \ge 0,$$

$$K_{*} = - \left(R + D^{T}PD\right)^{\dagger} \left(B^{T}P + D^{T}PC\right)$$

$$+ Y - \left(R + D^{T}PD\right)^{\dagger} \left(R + D^{T}PD\right)Y,$$
(35)

where  $Y \in L^2(0, T; \mathbb{R}^{m \times n})$ .

*Proof.* Because of Problem 21 without the equality constraint,  $\lambda = 0$ , therefore P(T) = 0. Theorem 20 yields this corollary directly.

#### 4. Application

4.1. A Necessary and Sufficient Condition. In Theorem 15, to ensure the uniqueness of  $\lambda$  and P, let us strengthen the condition  $(R + D^T PD) \ge 0$  as  $(R + D^T PD) > 0$ .

**Theorem 23.** If  $K_*$  is optimal solution of Problem 1, then  $P \in NBV^{n \times n}[0, T]$  and  $\lambda \in R^1 > 0$ , such that

$$-\dot{P} = \left(Q + K_*^T R K_*\right) + P \left(A + B K_*\right) + \left(A + B K_*\right)^T P + \left(C + D K_*\right) P \left(C + D K_*\right)^T, P (T) = \lambda I_n,$$
(36)  
$$K_*^T R + P B + \left(C + D K_*\right)^T P D = 0,$$

$$\lambda [\operatorname{tr} X(T) - c] = 0,$$

and if  $(R + D^T PD) > 0$  hold, then Problem 1 is well posed. The optimal solutions of Problem 1 are

$$U_{*} = -(R + D^{T}PD)^{-1} (B^{T}P + D^{T}PC) x,$$

$$J_{*} = x_{0}^{T}P(0) x_{0} - \lambda c.$$
(37)

*Proof.* In Remark 3, in order for a minimum point  $x^*$  of a mathematical programming to satisfy the KKT conditions, we have assumed that  $x^*$  satisfies LICQ regularity condition. From (19c) of Theorem 15, we obtain

$$K_*\left(R+D^T P D\right) = -\left(B^T P + D^T P C\right).$$
(38)

Because  $(R + D^T PD) > 0$ , Problem 11 has a unique solution; namely,

$$K_* = -\left(R + D^T P D\right)^{-1} \left(B^T P + D^T P C\right).$$
(39)

From the relationship between Problems 1 and 11 and by using Itô's formula, we have

$$J_{*} = V(x_{0}) = x_{0}^{T} P(0) x_{0} - \lambda c.$$
(40)

4.2. A Dynamic Programming Algorithm. A method for solving the indefinite stochastic linear quadratic (LQ) optimal control problem with unequal terminal state constraint is proposed as follows.

Reconsider the following LQ Problem 1'

$$dx(t) = [A(t) x(t) + B(t) u(t)] dt + [C(t) x(t) + D(t) u(t)] dw(t), x(s) = y, E {||x(T)||2} = E {xT(T) x(T)} \le c, J(x_0, u) = E { \int_0^T [x^T(t) Q(t) x(t) + u^T(t) R(t) u(t)] } dt, V(s, y) = \inf_{u(\cdot) \in U_{ad}} J(s, y; u(\cdot)).$$
(41)

Then the backward dynamic programming optimality principle [8] yields the following theorem.

**Theorem 24.** Let  $x_*(\cdot)$  and  $u_*(\cdot)$  be an optimal solution of Problem 1', then  $x_*(\cdot)$ , and  $u_*(\cdot)$  satisfy the following Bellman equation:

$$V(s, y) = \min_{u(\cdot) \in U_{ad}} E\left\{ \int_{s}^{s+h} \left[ x_{*}^{T}(t) Q(t) x_{*}(t) + u_{*}^{T}(t) R(t) u_{*}(t) \right] dt + V(s+h, x_{*}(s+h)) \right\} \quad \forall h > 0,$$
$$V(T-h, y) = \min_{u(\cdot) \in U_{ad}} E\left\{ \int_{T-h}^{T} \left[ x_{*}^{T}(t) Q(t) x_{*}(t) + u_{*}^{T}(t) R(t) u_{*}(t) \right] \right\} dt,$$
$$E\left\{ \| x(T) \|^{2} \right\} = E\left[ x^{T}(T) x(T) \right] \le c.$$
(42)

*Proof.* By applying the backward dynamic programming optimality principle, we obtain recursive relationships between  $V(s, x_*(s))$  and  $V(s + h, x_*(s + h))$  as follows:

$$V(s, y) = \min_{u(\cdot) \in U_{ad}} E\left\{ \int_{s}^{s+h} \left[ x_{*}^{T}(t) Q(t) x_{*}(t) + u_{*}^{T}(t) R(t) u_{*}(t) \right] dt + V(s+h, x_{*}(s+h)) \right\},$$
(43)

where  $V(s, x_*(s))$  is the objective of [s, T] and  $V(s + h, x_*(s + h))$  of [s + h, T]. Let  $V(s, x_*(s))$  be the objective of [T - h, T] and then

$$V(T - h, y) = \min_{u(\cdot) \in U_{ad}} E \int_{T-h}^{T} \left[ x_*^T(t) Q(t) x_*(t) + u_*^T(t) R(t) u_*(t) \right] dt.$$
(44)

Combine the terminal state constraint

$$E\left\{\left\|x\left(T\right)\right\|^{2}\right\} = E\left[x^{T}\left(T\right)x\left(T\right)\right] \le c,$$

$$u\left(t\right) = Kx\left(t\right) \quad \left(K \text{ is given by Theorem 20}\right);$$
(45)

thus we can solve the above recursive relations. Furthermore, let  $h \rightarrow 0$  generating the dynamic programming algorithm.

*Remark 25.* The dynamic programming algorithm can be applied to solve the stochastic LQ problem with inequality constraint except for a heavy computation. Nonetheless, it makes more sense in theory.

# *Example 26.* In Problem 1, let A = -1, B

$$A = -1, \qquad B = 1, \qquad C = -2,$$
  

$$D = 0, \qquad c = \frac{\left(1 + e^2\right)^2}{4e^2}, \qquad (46)$$
  

$$x_0 = 1, \qquad Q = 0,$$
  

$$R = 1, \qquad T = 1.$$

By Theorem 15, we have

$$-\dot{P} = K_*^2 + 2P(-1 + K_*) + 4P,$$

$$P(1) = \lambda,$$

$$K_* + P = 0,$$

$$\lambda \left[ X(1) - \frac{(1 + e^2)^2}{4e^2} \right] = 0,$$

$$\lambda \ge 0.$$
(47)

The constraints of Problem 11 are

$$-\dot{X} = 2X + 2KX,$$

$$X(0) = 1,$$

$$X(1) \le \frac{(1+e^2)^2}{4e^{2t}}.$$
(48)

Via a series of computations, we obtain

$$X_* = \frac{\left(1+e^2\right)^2}{4e^{2t}}, \qquad P = \frac{2}{1+e^{2t}},$$

$$\lambda = \frac{2}{1+e^2}, \qquad K_* = -\frac{2}{1+e^{2t}}.$$
(49)

In Problem 11, the inequality constraint is

$$G(X_*(1)) = X_*(1) - c \le 0.$$
(50)

When T = 1,

$$X_*(1) = \frac{(1+e^2)^2}{4e^2} = c.$$
 (51)

Hence

$$G(X_*(1)) = X_*(1) - c = 0.$$
(52)

This shows that the optimal solution of Problem 11 satisfies the inequality constraint. In this case, the inequality constraint is a nonactive constraint. Because  $(R + D^T PD) = 1 > 0$ , using Theorem 23, the optimal control of Problem 1 is

$$u_{*}(t) = -\frac{2}{1+e^{2t}}x_{*}(t), \qquad (53)$$

and the optimal value is

$$V(x_0) = x_0^T P(0) x_0 - \lambda c = \frac{1 - e^{-2}}{2}.$$
 (54)

# **5.** Conclusion

We have studied the indefinite stochastic LQ optimal control problem with unequal terminal state constraint, which can be transformed into a hybrid constrained mathematical programming problem. By applying KKT theorem, we have presented a necessary condition for the constrained indefinite stochastic LQ optimal control problem. By adding some conditions, we obtain a necessary and sufficient condition for indefinite stochastic LQ optimal control problem with inequality constraint. We demonstrate that the solvability of the generalized Riccati equation is sufficient for the well posedness of the indefinite LQ problem. Since this kind of LQ control problems can be transformed into a hybrid constrained mathematical programming, we have given a dynamic programming algorithm.

#### Acknowledgments

This work is supported by the National Natural Science Foundation of China (61174078, 61170183), the Research Fund for the Taishan Scholar Project of Shandong Province of China, the SDUST Research Fund (no. 2011KYTD105), and the State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources (no. LAPS13018).

#### References

- R. E. Kalman, "Contributions to the theory of optimal control," Boletín de la Sociedad Matemática Mexicana, vol. 5, no. 2, pp. 102–119, 1960.
- [2] W. M. Wonham, "On a matrix Riccati equation of stochastic control," *SIAM Journal on Control and Optimization*, vol. 6, pp. 681–697, 1968.
- [3] C. E. de Souza and M. D. Fragoso, "On the existence of maximal solution for generalized algebraic Riccati equations arising in stochastic control," *Systems & Control Letters*, vol. 14, no. 3, pp. 233–239, 1990.
- [4] W. Zhang, "Study on generalized algebraic Riccati equation and optimal regulators," *Journal of Control Theory and Applications*, vol. 20, no. 4, pp. 637–640, 2003.
- [5] S. Chen, X. Li, and X. Y. Zhou, "Stochastic linear quadratic regulators with indefinite control weight costs," *SIAM Journal* on Control and Optimization, vol. 45, no. 6, pp. 1131–1143, 2000.
- [6] M. A. Rami and X. Y. Zhou, "Linear matrix inequalities, Riccati equations, and indefinite stochastic linear quadratic controls," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1131– 1143, 2000.
- [7] C. E. de Souza and M. D. Fragoso, "On the existence of maximal solution for generalized algebraic Riccati equations arising in stochastic control," *Systems & Control Letters*, vol. 14, no. 3, pp. 233–239, 1990.
- [8] R. Bellman, "On the theory of dynamic programming," Proceedings of the National Academy of Sciences of the United States of America, vol. 38, pp. 716–719, 1952.
- [9] D. D. Yao, S. Zhang, and X. Y. Zhou, "Stochastic linear-quadratic control via semidefinite programming," *SIAM Journal on Control and Optimization*, vol. 40, no. 3, p. 801–823 (electronic), 2001.

- [10] W. Zhang and B.-S. Chen, "On stabilizability and exact observability of stochastic systems with their applications," *Automatica*, vol. 40, no. 1, pp. 87–94, 2004.
- [11] A. Beghi and D. D'Alessandro, "Discrete-time optimal control with control-dependent noise and generalized Riccati difference equations," *Automatica*, vol. 34, no. 8, pp. 1031–1034, 1998.
- [12] D. D. Yao, S. Zhang, and X. Y. Zhou, "Stochastic linearquadratic control via semidefinite programming," *SIAM Journal* on Control and Optimization, vol. 40, no. 3, pp. 801–823, 2001.
- [13] M. Ait Rami, J. B. Moore, and X. Y. Zhou, "Indefinite stochastic linear quadratic control and generalized differential Riccati equation," *SIAM Journal on Control and Optimization*, vol. 40, no. 4, pp. 1296–1311, 2001.
- [14] M. Ait Rami, X. Chen, and X. Y. Zhou, "Discrete-time indefinite LQ control with state and control dependent noises," *Journal of Global Optimization*, vol. 23, no. 3-4, pp. 245–265, 2002.
- [15] Y. Huang, W. Zhang, and H. Zhang, "Infinite horizon linear quadratic optimal control for discrete-time stochastic systems," *Asian Journal of Control*, vol. 10, no. 5, pp. 608–615, 2008.
- [16] G. Wang and Z. Wu, "Stochastic linear quadratic optimal control with partial information and its application to meanvariance hedging problems," in *Proceedings of the 27th Chinese Control Conference (CCC '08)*, pp. 601–605, Yunnan, China, July 2008.
- [17] Y. L. Huang and W. H. Zhang, "Study on stochastic linear quadratic optimal control with constraint," *Acta Automatica Sinica*, vol. 32, no. 2, pp. 246–254, 2006.
- [18] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492, Berkeley University, California Press, 1951.
- [19] W. Karush, Minima of functions of several variables with inequalities as side constraints [M.S. thesis], Department of Mathematics, University of Chicago, Chicago, Ill, USA, 1939.
- [20] T. H. Kjeldsen, "A contextualized historical analysis of the Kuhn-Tucker theorem in nonlinear programming: the impact of World War II," *Historia Mathematica*, vol. 27, no. 4, pp. 331– 361, 2000.
- [21] D. G. Luenberger, Optimization by Vector Space Methods, John Wiley & Sons, New York, NY, USA, 1968.
- [22] M. Athans, "The matrix minimum principle," *Information and Computation*, vol. 11, pp. 592–606, 1967.

# **Research** Article

# On Iterative Learning Control for Remote Control Systems with Packet Losses

# Chunping Liu,<sup>1,2</sup> Rong Xiong,<sup>1</sup> Jianxin Xu,<sup>1,3</sup> and Jun Wu<sup>1</sup>

<sup>1</sup> State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

<sup>2</sup> Department of Electric and Information Engineering, Shaoxing College of Arts and Sciences, Shaoxing 31200, China

<sup>3</sup> Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576

Correspondence should be addressed to Rong Xiong; rxiong@iipc.zju.edu.cn

Received 16 May 2013; Accepted 8 October 2013

Academic Editor: Neal N. Xiong

Copyright © 2013 Chunping Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The problem of iterative learning control (ILC) is considered for a class of time-varying systems with random packet dropouts. It is assumed that an ILC scheme is implemented via a remote control system and that packet dropout occurs during the packet transmission between the ILC controller and the actuator of remote plant. The packet dropout is viewed as a binary switching sequence which is subject to the Bernoulli distribution. In order to eliminate the effect of packet dropouts on the convergence property of output error, the hold-input scheme is adopted to compensate the packet dropout at the actuator. It is shown that the hold-input scheme with average ILC can achieve asymptotical convergence along the iteration axis for discrete time-varying linear system. Numerical examples are provided to show the effectiveness of the proposed method.

# 1. Introduction

Iterative learning control (ILC) is an attractive technique when dealing with systems that execute the same task repeatedly over a finite time interval [1]. This technique has been the center of interest of many researchers over the two decades [2–5] and covered a wide scope of research issues such as model uncertainty [6–8], disturbance uncertainty and stochastic noise [9], the initial condition and desired trajectory uncertainty [10–12], continuous-time nonlinear system control [13], and parameter interval uncertainty [14].

On the other hand, the remote control systems have been the focus of several research studies over the last few years [15–21]. In the remote control systems, one feature is that the control loops are closed through a real-time communication channel which transmits signals from the sensors to the controller and from the controller to the actuators [17]. The remote control systems eliminate unnecessary wiring reducing the complexity and overall cost in designing and implementing the control systems. However, the introduction of communication networks makes the analysis and control design more complicated than classical feedback loops. Data packet dropout can randomly occur due to node failure or network congestion and impose one of the most important issues in remote control systems. In [18, 19], the authors are concerned with the stability problem for remote control systems with the packet dropout. In the work [20, 21], decentralized stabilization of remote control systems with nonlinear perturbations is studied.

Besides the stability issue, trajectory tracking is a challenging issue for remote control systems. Fortunately, for periodic systems, iterative learning control offers a systematic design that can improve the tracking performance by iterations in a fixed time interval. ILC is in principle a feedforward technique; thus it can send the controller signals obtained from previous trials. It is still an open research area in ILC which is implemented via a remote systems setting, except for certain pioneer works [22–29]. In [22, 23], the authors designed an optimal ILC controller for a class of linear systems with random packet dropouts. Bu et al. [26] studied the stability of first and high order ILC with data dropout when the plant is subject to measurement signal dropout. In [24, 25], the authors investigated the implementation of ILC in a remote control systems environment and specifically focused on compensation when both random data dropouts and delays occur at the communication network between the sensors and the controller. In [27], a sampled-data ILC approach was proposed for a class of nonlinear remote control systems to analyze the effect of packet loss. In [28], the author considered the problem of ILC for a class of nonlinear systems with control signal dropouts and measurement signal dropouts, but the convergence analysis needs controller and actuator to know the received signal whether lost or not. Huang and Fang [29] discussed the wireless remote iterative learning control system with random data dropouts.

In this paper, we proposed an ILC for a time-varying system with random packet dropouts. As depicted in previous studies [22-29], there are two different kinds of packet dropouts in remote ILC systems: control input signal dropouts and output measurement signal dropouts. For the sake of convenience, we only consider the control signal dropouts in this paper, but the results can be extended to the measurement signal dropouts. The packet dropouts would be described as a binary sequence which is subject to a Bernoulli distribution taking the value of one or zero with certain probability. The ILC law adopts an iteration-average operator and a revised learning gain that takes into consideration the probabilities of data-dropout factors. As a result, the ensemble average of the output tracking errors can be made to converge along the iteration axis. In this paper, we consider a class of discrete time linear plants with output matrix C and input matrix **B**; our results refer only to **CB** of full-column rank.

The paper is organized as follows. Section 2 formulates the system problem. Section 3 formulates the hold-input scheme with average ILC algorithm and proves the convergence property of ILC for linear varying discrete-time plants. Section 4 presents numerical examples, and Section 5 draws the conclusions.

#### 2. Problem Formulation

Consider the ILC system with network communication depicted in Figure 1. The discrete time linear plant with actuators and sensors is described by

$$\mathbf{x}_{i}(t+1) = \mathbf{A}(t) \mathbf{x}_{i}(t) + \mathbf{B}(\mathbf{u}_{i}(t) + \mathbf{w}_{i}(t))$$
  

$$\mathbf{y}_{i}(t) = \mathbf{C}\mathbf{x}_{i}(t) + \mathbf{v}_{i}(t),$$
(1)

where  $i \in \mathbb{Z}_+$  denotes the iteration index;  $T \in \mathbb{Z}_+$  is a given finite time;  $\mathbf{x}_i(t) \in \mathbb{R}^n$ ,  $\mathbf{u}_i(t) \in \mathbb{R}^q$ , and  $\mathbf{y}_i(t) \in \mathbb{R}^m$  are state, control, and output, respectively;  $\mathbf{A}(t) \in \mathbb{R}^{n \times n}$  is unknown matrix, while  $\mathbf{B} \in \mathbb{R}^{n \times q}$  and  $\mathbf{C} \in \mathbb{R}^{m \times n}$  are known;  $\mathbf{w}_i(t) \in \mathbb{R}^q$ and  $\mathbf{v}_i(t) \in \mathbb{R}^m$  are random noises with  $\mathscr{E}[\mathbf{w}_i(t)] = \mathbf{0}$  and  $\mathscr{E}[\mathbf{v}_i(t)] = \mathbf{0}$ ; for all  $i \in \mathbb{Z}_+$ , the initial state  $\mathbf{x}_i(0)$  is a random variable of  $\mathscr{E}[\mathbf{x}_i(0)] = \mathbf{x}_0$  with a fixed point  $\mathbf{x}_0 \in \mathbb{R}^n$ . Assume that **CB** has full-column rank. The discrete time controller consists of a ILC algorithm and a memory. The controller and the actuators are connected via a communication network through which the controller transmits data to the actuators, while the controller is directly connected to the sensors. The plant and the controller are assumed to be time driven and synchronized.

At each  $t \in \{0, ..., T\}$  of the *i*th iteration stage, the controller output  $\hat{\mathbf{u}}_i(t)$  is computed, the controller transmits  $\hat{\mathbf{u}}_i(t)$  to the actuators through the network. The transmission may succeed or fail. For a successful transmission, it is assumed that the transmission delay through the network is negligible. With the negligible delay, the actuators can employ  $\mathbf{u}_i(t) = \hat{\mathbf{u}}_i(t)$ , when  $\hat{\mathbf{u}}_i(t)$  is transmitted successfully. Of course, when the transmission fails, the actuators receive no  $\hat{\mathbf{u}}_i(t)$  and have to employ  $\mathbf{u}_i(t) = \mathbf{u}_i(t-1)$  (this paper prescribes  $\mathbf{u}_i(-1) = 0$ ). Overall, the scheme of actuators is

$$\mathbf{u}_{i}(t) = \gamma_{i}(t)\,\widehat{\mathbf{u}}_{i}(t) + (1 - \gamma_{i}(t))\,\mathbf{u}_{i}(t - 1)\,,\tag{2}$$

where

$$\gamma_i(t) = \begin{cases} 1, & \text{if the transmission of } \widehat{\mathbf{u}}_i(t) \text{ succeeds,} \\ 0, & \text{if the transmission of } \widehat{\mathbf{u}}_i(t) \text{ fails.} \end{cases}$$
(3)

Specially, this paper assumes that, for all  $i \in \mathbb{Z}_+$ , for all  $t \in \{0, ..., T\}$ ,  $\gamma_i(t)$  is a random variable of  $\mathscr{E}[\gamma_i(t)] = \gamma$  with a constant  $\gamma \in (0, 1)$  as well as that  $\gamma_i(t_1)$  and  $\gamma_j(t_2)$  are independent either when  $i \neq j$  or when  $t_1 \neq t_2$ . In addition, TCP-like protocol is assumed, in which there is an acknowledgment for a successful transmission, and hence the controller has indicators of whether the current controller output is received or not by the actuators.

Assumption 1. Given an output reference trajectory  $\mathbf{y}_d(t)$ , which is realizable; that is, there exists a unique desired control input  $\mathbf{u}_d(t) \in \mathbb{R}^q$  such that

$$\mathbf{x}_{d}(t+1) = \mathbf{A}(t)\mathbf{x}_{d}(t) + \mathbf{B}\mathbf{u}_{d}(t)$$
$$\mathbf{x}_{d}(0) = \mathbf{x}_{0}. \quad (4)$$
$$\mathbf{y}_{d}(t) = \mathbf{C}\mathbf{x}_{d}(t),$$

The purpose of this paper is to design an iterative learning control law for the above plant with network communication such that  $\mathbf{y}_i(t)$  tracks  $\mathbf{y}_d(t)$  as closely as possible when *i* is large enough.

#### 3. ILC Algorithms and Convergence Analysis

Denote  $\mathbf{e}_i(t) \triangleq \mathbf{y}_d(t) - \mathbf{y}_i(t)$ . The control law is a D-type ILC with average operator that employs updating mechanism:

$$\widehat{\mathbf{u}}_{i+1}(t) = \frac{1}{i} \sum_{j=1}^{i} \mathbf{u}_{j}(t) + \frac{i+1}{i} \mathbf{L} \sum_{j=1}^{i} \mathbf{e}_{j}(t+1)$$

$$= \mathscr{A} \left[ \mathbf{u}_{i}(t) \right] + (i+1) \mathbf{L} \mathscr{A} \left[ \mathbf{e}_{i}(t+1) \right],$$
(5)

where the gain matrix  $\mathbf{L} \in \mathbb{R}^{q \times m}$ . From (2) and (5), the hold-input scheme with average ILC is expressed as

$$\mathbf{u}_{i+1}(t) = \gamma_{i+1}(t) \, \widehat{\mathbf{u}}_{i+1}(t) + (1 - \gamma_{i+1}(t)) \, \mathbf{u}_{i+1}(t - 1)$$
  
=  $\gamma_{i+1}(t) \left( \mathscr{A} \left[ \mathbf{u}_{i}(t) \right] + (i + 1) \, \mathbf{L} \mathscr{A} \left[ \mathbf{e}_{i}(t + 1) \right] \right)$   
+  $\left( 1 - \gamma_{i+1}(t) \right) \mathbf{u}_{i+1}(t - 1) \, .$  (6)



FIGURE 1: The schematic diagram of the networked control system.

Define the input and state errors

$$\Delta \mathbf{u}_{i+1}(t) \triangleq \mathbf{u}_{d}(t) - \mathbf{u}_{i+1}(t),$$

$$\Delta \mathbf{x}_{i+1}(t) \triangleq \mathbf{x}_{d}(t) - \mathbf{x}_{i+1}(t).$$
(7)

And subtracting  $\mathbf{u}_d(t)$  from both sides of (6) yields

$$\Delta \mathbf{u}_{i+1}(t) = \gamma_{i+1}(t) \left( \mathscr{A} \left[ \Delta \mathbf{u}_{i}(t) \right] - (i+1) \mathbf{L} \mathscr{A} \left[ \mathbf{e}_{i}(t+1) \right] \right) + \left( 1 - \gamma_{i+1}(t) \right) \left( \Delta \mathbf{u}_{i+1}(t-1) + \boldsymbol{\delta}(t) \right),$$
(8)

where (this paper prescribes  $\mathbf{u}_d(-1) = 0$  and hence  $\boldsymbol{\delta}(0) = \mathbf{u}_d(0)$ )  $\boldsymbol{\delta}(t) \triangleq \mathbf{u}_d(t) - \mathbf{u}_d(t-1)$ . Noticing that  $\gamma_{i+1}(t)$  is independent of  $\mathscr{A}[\Delta \mathbf{u}_i(t)]$ ,  $\mathscr{A}[\mathbf{e}_i(t+1)]$  and  $\Delta \mathbf{u}_{i+1}(t-1)$  and taking expectation on both sides of (8), we have

$$\mathscr{E}\left[\Delta \mathbf{u}_{i+1}\left(t\right)\right]$$

$$= \gamma\left(\mathscr{E}\left[\mathscr{A}\left[\Delta \mathbf{u}_{i}\left(t\right)\right]\right] - (i+1) \mathbf{L}\mathscr{E}\left[\mathscr{A}\left[\mathbf{e}_{i}\left(t+1\right)\right]\right]\right)$$

$$+ \left(1-\gamma\right)\left(\mathscr{E}\left[\Delta \mathbf{u}_{i+1}\left(t-1\right)\right] + \boldsymbol{\delta}\left(t\right)\right).$$
(9)

Expanding expression (9) from  $\mathscr{C}[\Delta \mathbf{u}_{i+1}(t-1)]$  to  $\mathscr{C}[\Delta \mathbf{u}_{i+1}(0)]$ , we have

$$\mathscr{C} \left[ \Delta \mathbf{u}_{i+1} \left( t - 1 \right) \right]$$
  
=  $\gamma \left( \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( t - 1 \right) \right] \right] - \left( i + 1 \right) \mathbf{L} \mathscr{C} \left[ \mathscr{A} \left[ \mathbf{e}_{i} \left( t \right) \right] \right] \right)$   
+  $\left( 1 - \gamma \right) \left( \mathscr{C} \left[ \Delta \mathbf{u}_{i+1} \left( t - 2 \right) \right] + \boldsymbol{\delta} \left( t - 1 \right) \right),$ 

$$\mathscr{E} \left[ \Delta \mathbf{u}_{i+1} \left( t - 2 \right) \right]$$
  
=  $\gamma \left( \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( t - 2 \right) \right] \right] - (i+1) \mathbf{L} \mathscr{E} \left[ \mathscr{A} \left[ \mathbf{e}_{i} \left( t - 1 \right) \right] \right] \right)$   
+  $\left( 1 - \gamma \right) \left( \mathscr{E} \left[ \Delta \mathbf{u}_{i+1} \left( t - 3 \right) \right] + \boldsymbol{\delta} \left( t - 2 \right) \right)$   
:

$$\mathscr{E} \left[ \Delta \mathbf{u}_{i+1} \left( 0 \right) \right]$$

$$= \gamma \left( \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( 0 \right) \right] \right] - \left( i+1 \right) \mathbf{L} \mathscr{E} \left[ \mathscr{A} \left[ \mathbf{e}_{i} \left( 1 \right) \right] \right] \right)$$

$$+ \left( 1-\gamma \right) \left( \mathscr{E} \left[ \Delta \mathbf{u}_{i+1} \left( -1 \right) \right] + \boldsymbol{\delta} \left( 0 \right) \right)$$

$$= \gamma \left( \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( 0 \right) \right] \right] - \left( i+1 \right) \mathbf{L} \mathscr{E} \left[ \mathscr{A} \left[ \mathbf{e}_{i} \left( 1 \right) \right] \right] \right)$$

$$+ \left( 1-\gamma \right) \boldsymbol{\delta} \left( 0 \right).$$
(10)

The above expression can be arranged later below (this paper prescribes  $\sum_{k=k_1}^{k_2}=0$  when  $k_2 < k_1)$ 

$$\mathscr{C}\left[\Delta \mathbf{u}_{i+1}\left(t-1\right)\right] = \sum_{k=0}^{t-1} \gamma (1-\gamma)^{k} \mathscr{C}\left[\mathscr{A}\left[\Delta \mathbf{u}_{i}\left(t-k-1\right)\right]\right]$$
$$-\left(i+1\right) \mathbf{L} \sum_{k=0}^{t-1} \gamma (1-\gamma)^{k} \mathscr{C}\left[\mathscr{A}\left[\mathbf{e}_{i}\left(t-k\right)\right]\right]$$
$$+ \sum_{k=0}^{t-1} (1-\gamma)^{k+1} \boldsymbol{\delta}\left(t-k-1\right).$$
(11)

From (1) and (4), we have

$$\Delta \mathbf{x}_{i} (t+1) = \mathbf{A} (t) \Delta \mathbf{x}_{i} (t) + \mathbf{B} (\Delta \mathbf{u}_{i} (t) - \mathbf{w}_{i} (t))$$
  
$$\mathbf{e}_{i} (t) = \mathbf{C} \Delta \mathbf{x}_{i} (t) - \mathbf{v}_{i} (t).$$
 (12)

Taking expectation on both sides of (12) and expanding expression from  $\mathscr{C}[\Delta \mathbf{x}_i(t+1)]$  to  $\mathscr{C}[\Delta \mathbf{x}_i(1)]$ , we obtain

$$\mathscr{E} \left[ \Delta \mathbf{x}_{i} \left( t+1 \right) \right] = \mathbf{A} \left( t \right) \mathscr{E} \left[ \Delta \mathbf{x}_{i} \left( t \right) \right] + \mathbf{B} \mathscr{E} \left[ \Delta \mathbf{u}_{i} \left( t \right) \right],$$
  

$$\mathscr{E} \left[ \Delta \mathbf{x}_{i} \left( t \right) \right] = \mathbf{A} \left( t-1 \right) \mathscr{E} \left[ \Delta \mathbf{x}_{i} \left( t-1 \right) \right]$$
  

$$+ \mathbf{B} \mathscr{E} \left[ \Delta \mathbf{u}_{i} \left( t-1 \right) \right],$$
  

$$\vdots$$
  

$$\mathscr{E} \left[ \Delta \mathbf{x}_{i} \left( 1 \right) \right] = \mathbf{A} \left( 0 \right) \mathscr{E} \left[ \Delta \mathbf{x}_{i} \left( 0 \right) \right] + \mathbf{B} \mathscr{E} \left[ \Delta \mathbf{u}_{i} \left( 0 \right) \right].$$
(13)

The above expression can be arranged later (this paper prescribes  $\prod_{\tau=\tau_1}^{\tau_2} = \mathbf{I}$  when  $\tau_2 < \tau_1$ )

$$\mathscr{C}\left[\Delta \mathbf{x}_{i}\left(t+1\right)\right] = \sum_{\tau=0}^{t} \prod_{\nu=\tau+1}^{t} \mathbf{A}\left(\nu\right) \mathbf{B}\mathscr{C}\left[\Delta \mathbf{u}_{i}\left(\tau\right)\right]$$
$$\mathscr{C}\left[\mathscr{A}\left[\mathbf{e}_{i}\left(t+1\right)\right]\right] = \mathbf{C}\mathscr{C}\left[\mathscr{A}\left[\Delta \mathbf{x}_{i}\left(t+1\right)\right]\right]$$
$$= \mathbf{C}\sum_{\tau=0}^{t} \prod_{\nu=\tau+1}^{t} \mathbf{A}\left(\nu\right) \mathbf{B}\mathscr{C}\left[\mathscr{A}\left[\Delta \mathbf{u}_{i}\left(\tau\right)\right]\right].$$
(14)

For any a > 1 and any  $\lambda > 1$ , denote

$$\left\| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \right] \right] \right\|_{(\lambda,a)} \triangleq \max_{t \in \{0,1,\dots,T\}} a^{-\lambda t} \left\| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( t \right) \right] \right] \right\|_{2},$$
(15)

$$\left\| \mathscr{C} \left[ \mathscr{A} \left[ \mathbf{e}_{i} \right] \right] \right\|_{(\lambda,a)} \triangleq \max_{t \in \{0,1,\dots,T\}} a^{-\lambda t} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \mathbf{e}_{i} \left( t \right) \right] \right] \right\|_{2}.$$
(16)

**Lemma 2.** For all a > 1, for all  $\lambda > 1$ , and for all  $i \in \mathbb{Z}_+$ ,

$$\gamma \max_{t \in \{0,\dots,T\}} a^{-\lambda t} \sum_{k=0}^{t} (1-\gamma)^{k} \sum_{\tau=0}^{t-k-1} a^{t-k-\tau} \| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( \tau \right) \right] \right] \|_{2}$$
$$\leq \frac{\| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \right] \right] \|_{(\lambda,a)}}{a^{\lambda-1}-1}.$$
(17)

Proof. From (17), we have

$$\begin{split} \gamma \max_{t \in \{0,...,T\}} a^{-\lambda t} \sum_{k=0}^{t} (1-\gamma)^{k} \\ & \times \sum_{\tau=0}^{t-k-1} a^{t-k-\tau} \| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( \tau \right) \right] \right] \|_{2} \\ & \leq \gamma \max_{t \in \{0,...,T\}} \sum_{k=0}^{t} (1-\gamma)^{k} \\ & \times \sum_{\tau=0}^{t-k-1} a^{-\lambda \tau} \| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( \tau \right) \right] \right] \|_{2} a^{-(\lambda-1)t} a^{(\lambda-1)\tau} \end{split}$$

$$\leq \gamma \max_{t \in \{0,...,T\}} \sum_{k=0}^{t} (1-\gamma)^{k} \| \mathscr{C} [\mathscr{A} [\Delta \mathbf{u}_{i}]] \|_{(\lambda,a)}$$

$$\times \sum_{\tau=0}^{t-k-1} a^{-(\lambda-1)t} a^{(\lambda-1)\tau}$$

$$\leq \| \mathscr{C} [\mathscr{A} [\Delta \mathbf{u}_{i}]] \|_{(\lambda,a)} \max_{t \in \{0,...,T\}} \gamma$$

$$\times \sum_{k=0}^{t} (1-\gamma)^{k} \frac{a^{-(\lambda-1)k} - a^{-(\lambda-1)t}}{a^{\lambda-1} - 1}$$

$$\leq \frac{\| \mathscr{C} [\mathscr{A} [\Delta \mathbf{u}_{i}]] \|_{(\lambda,a)}}{a^{\lambda-1} - 1} \max_{t \in \{0,...,T\}} \gamma \frac{1 - (1-\gamma)^{t+1}}{1 - (1-\gamma)}$$

$$\leq \frac{\| \mathscr{C} [\mathscr{A} [\Delta \mathbf{u}_{i}]] \|_{(\lambda,a)}}{a^{\lambda-1} - 1}.$$
(18)

**Theorem 3.** For the system with network communication described in Section 2 and the iterative learning controller (5), suppose

$$\rho \triangleq \|\mathbf{I} - \gamma \mathbf{LCB}\| < 1. \tag{19}$$

Then for all  $\mathbf{A}_0, \ldots, \mathbf{A}_T \in \mathbb{R}^{n \times n}$ , for all  $\epsilon > 0$ , there exist a > 1 and  $\lambda > 1$  such that

$$\lim_{i \to \infty} \|\mathscr{E}\left[\mathbf{e}_{i}\right]\|_{(\lambda,a)} < \epsilon.$$
(20)

Proof. From definition of average operator, note the relation

$$\mathscr{A}\left[\Delta \mathbf{u}_{i+1}\left(t\right)\right] = \frac{1}{i+1} \left(\Delta \mathbf{u}_{i+1}\left(t\right) + \sum_{j=1}^{i} \Delta \mathbf{u}_{j}\left(t\right)\right)$$
$$= \frac{1}{i+1} \left(\Delta \mathbf{u}_{i+1}\left(t\right) + i\mathscr{A}\left[\Delta \mathbf{u}_{i}\left(t\right)\right]\right).$$
(21)

Applying the ensemble operator  $\mathscr{C}[\cdot]$  to both sides of (21) and substituting the relationship (8), we can obtain

$$\mathscr{E}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i+1}\left(t\right)\right]\right]$$

$$=\frac{1}{i+1}\mathscr{E}\left[\Delta\mathbf{u}_{i+1}\left(t\right)\right] + \frac{i}{i+1}\mathscr{E}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right]$$

$$=\frac{\gamma}{i+1}\left(\mathscr{E}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right] - (i+1)\mathbf{L}\mathscr{E}\left[\mathscr{A}\left[\mathbf{e}_{i}\left(t+1\right)\right]\right]\right)$$

$$+\frac{1-\gamma}{i+1}\left(\mathscr{E}\left[\Delta\mathbf{u}_{i+1}\left(t-1\right)\right] + \boldsymbol{\delta}\left(t\right)\right)$$

$$+\frac{i}{i+1}\mathscr{E}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right].$$
(22)

Substituting (11) into (22) leads to the following relationship:

$$\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i+1}\left(t\right)\right]\right] = \frac{i+\gamma}{i+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right] - \gamma \mathbf{L}\mathscr{C}\left[\mathscr{A}\left[\mathbf{e}_{i}\left(t+1\right)\right]\right] + \frac{\gamma}{i+1}\sum_{k=0}^{t-1}(1-\gamma)^{k+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t-k-1\right)\right]\right] - \gamma \mathbf{L}\sum_{k=0}^{t-1}(1-\gamma)^{k+1}\mathscr{C}\left[\mathscr{A}\left[\mathbf{e}_{i}\left(t-k\right)\right]\right] + \frac{1-\gamma}{i+1}\left(\sum_{k=0}^{t-1}(1-\gamma)^{k+1}\boldsymbol{\delta}\left(t-k-1\right)+\boldsymbol{\delta}\left(t\right)\right) = \frac{i+\gamma}{i+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right] + \frac{\gamma}{i+1} \times \sum_{k=0}^{t-1}(1-\gamma)^{k+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t-k-1\right)\right]\right] - \gamma \mathbf{L}\sum_{k=0}^{t}(1-\gamma)^{k+1}\mathscr{C}\left[\mathscr{A}\left[\mathbf{\Delta}\left[\mathbf{e}_{i}\left(t-k+1\right)\right]\right] + \frac{1}{i+1}\sum_{k=0}^{t}(1-\gamma)^{k+1}\boldsymbol{\delta}\left(t-k\right).$$

$$(23)$$

Now let us handle the third term on the right hand side of (23); we will express  $\mathscr{C}[\mathscr{A}[\mathbf{e}_i(t-k+1)]]$  with  $\mathscr{C}[\mathscr{A}[\mathbf{u}_i]]$ . Substituting the state error dynamics (14) into (23) leads to the following relationship:

$$\begin{split} \mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i+1}\left(t\right)\right]\right] \\ &= \frac{i+\gamma}{i+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right] + \frac{\gamma}{i+1} \\ &\times \sum_{k=0}^{t-1} (1-\gamma)^{k+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t-k-1\right)\right]\right] \\ &- \gamma \mathbf{LC}\sum_{k=0}^{t} (1-\gamma)^{k} \\ &\times \sum_{\tau=0}^{t-k} \prod_{\nu=\tau+1}^{t-k} \mathbf{A}\left(\nu\right) \mathbf{B}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(\tau\right)\right]\right] + \boldsymbol{\mu}_{i}\left(t\right) \\ &= \frac{i+\gamma}{i+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right] + \frac{\gamma}{i+1} \\ &\times \sum_{k=1}^{t} (1-\gamma)^{k}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t-k\right)\right]\right] \\ &- \gamma \mathbf{LC}\sum_{k=0}^{t} (1-\gamma)^{k} \end{split}$$

$$\times \sum_{\tau=0}^{t-k-1} \prod_{\nu=\tau+1}^{t-k} \mathbf{A}(\nu) \mathbf{B} \mathscr{C} \left[\mathscr{A} \left[\Delta \mathbf{u}_{i}(\tau)\right]\right]$$
$$-\gamma \mathbf{L} \mathbf{C} \sum_{k=0}^{t} (1-\gamma)^{k} \mathbf{B} \mathscr{C} \left[\mathscr{A} \left[\Delta \mathbf{u}_{i}(t-k)\right]\right] + \boldsymbol{\mu}_{i}(t), \qquad (24)$$

where  $\boldsymbol{\mu}_i(t) \triangleq (1/(i+1)) \sum_{k=0}^t (1-\gamma)^{k+1} \boldsymbol{\delta}(t-k)$ . Next, combining analogous terms on the right hand of (22), we obtain

$$\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i+1}\left(t\right)\right]\right]$$

$$=\frac{i+\gamma}{i+1}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right]+\frac{\gamma}{i+1}$$

$$\times\sum_{k=1}^{t}\left(1-\gamma\right)^{k}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t-k\right)\right]\right]$$

$$-\gamma\mathbf{LC}\sum_{k=0}^{t}\left(1-\gamma\right)^{k}\mathbf{B}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(\tau\right)\right]\right]$$

$$-\gamma\mathbf{LC}\sum_{k=1}^{t}\left(1-\gamma\right)^{k}\mathbf{B}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t-k\right)\right]\right]$$

$$-\gamma\mathbf{LC}\mathscr{B}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right]+\mu_{i}\left(t\right)$$

$$=\left(\frac{i+\gamma}{i+1}\mathbf{I}-\gamma\mathbf{LC}\mathbf{B}\right)\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right]$$

$$+\left(\frac{\gamma}{i+1}\mathbf{I}-\gamma\mathbf{LC}\mathbf{B}\right)\sum_{k=1}^{t}\left(1-\gamma\right)^{k}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t-k\right)\right]\right]$$

$$-\gamma\mathbf{LC}\sum_{k=0}^{t}\left(1-\gamma\right)^{k}$$

$$\times\sum_{\tau=0}^{t}\prod_{\nu=\tau+1}^{t-k}\mathbf{A}\left(\nu\right)\mathbf{B}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(\tau\right)\right]\right]+\mu_{i}\left(t\right).$$
(25)

The relationship (25) can be rewritten as follows:

$$\begin{aligned} \mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i+1}\left(t\right)\right]\right] \\ &= \left(\frac{i+\gamma}{i+1}\mathbf{I} - \gamma\mathbf{LCB}\right)\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(t\right)\right]\right] \\ &+ \left(\frac{\gamma}{i+1}\mathbf{I} - \gamma\mathbf{LCB}\right)\sum_{\tau=0}^{t-1}(1-\gamma)^{t-\tau}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(\tau\right)\right]\right] \end{aligned}$$

$$-\gamma \mathbf{LC} \sum_{k=0}^{t} (1-\gamma)^{k}$$

$$\times \sum_{\tau=0}^{t-k-1} \prod_{\nu=\tau+1}^{t-k} \mathbf{A}(\nu) \mathbf{B} \mathscr{C} \left[\mathscr{A} \left[\Delta \mathbf{u}_{i}(\tau)\right]\right] + \boldsymbol{\mu}_{i}(t) .$$
(26)

To simplify expression of  $((i+\gamma)/(i+1))\mathbf{I}-\gamma \mathbf{LCB}$ ,  $(\gamma/(i+1))\mathbf{I}-\gamma \mathbf{LCB}$ , and  $\gamma \mathbf{LC} \sum_{k=0}^{t} (1-\gamma)^{k}$ , we choose a > 1 and  $\lambda > 1$  such that

$$a > \|\mathbf{A}(\nu)\| \quad \forall \nu \in \{0, \dots, T\},$$
(27)

$$\frac{\gamma \|\mathbf{LCB}\| + \gamma + \|\mathbf{LC}\| \|\mathbf{B}\|}{a^{\lambda - 1} - 1} < \frac{1 - \rho}{4},$$
(28)

$$\|\mathbf{C}\| \|\mathbf{B}\| \frac{1}{a^{\lambda-1}-1} \frac{2\alpha}{1-\rho} < \epsilon,$$
(29)

where

$$\alpha \triangleq \max_{t \in \{0,...,T\}} \frac{a^{-\lambda t}}{2} \sum_{k=0}^{t} (1-\gamma)^{k+1} \|\boldsymbol{\delta}(t-k)\|_{2}$$

$$\geq \max_{t \in \{0,...,T\}} a^{-\lambda t} \|\boldsymbol{\mu}_{i}(t)\|_{2}.$$
(30)

Taking  $\lambda$ -norm on both sides of (26), we obtain

$$\begin{aligned} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i+1} \right] \right] \right\|_{(\lambda,a)} \\ &\leq \left\| \mathbf{I} - \gamma \mathbf{L} \mathbf{C} \mathbf{B} - \frac{1-\gamma}{i+1} \mathbf{I} \right\| \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \right] \right] \right\|_{(\lambda,a)} \\ &+ \max_{t \in \{0,...,T\}} a^{-\lambda t} \left\| \boldsymbol{\mu}_{i}\left( t \right) \right\|_{2} \\ &+ \left\| \frac{\gamma}{i+1} \mathbf{I} - \gamma \mathbf{L} \mathbf{C} \mathbf{B} \right\| \max_{t \in \{0,...,T\}} a^{-\lambda t} \\ &\times \sum_{\tau=0}^{t-1} (1-\gamma)^{t-\tau} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i}\left( \tau \right) \right] \right] \right\|_{2} \\ &+ \gamma \left\| \mathbf{L} \mathbf{C} \right\| \left\| \mathbf{B} \right\| \max_{t \in \{0,...,T\}} a^{-\lambda t} \\ &\times \sum_{k=0}^{t} (1-\gamma)^{k} \sum_{\tau=0}^{t-k-1} a^{t-k-\tau} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i}\left( \tau \right) \right] \right] \right\|_{2}. \end{aligned}$$

$$(31)$$

Using Lemma 2, it can be proved that, for all, a > 1, for all  $\lambda > 1$ , and for all  $i \in \mathbb{Z}_+$ ,

$$\max_{t \in \{0,...,T\}} a^{-\lambda t} \sum_{\tau=0}^{t-1} (1-\gamma)^{t-\tau} \| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( \tau \right) \right] \right] \|_{2}$$

$$\leq \frac{\| \mathscr{E} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \right] \right] \|_{(\lambda,a)}}{a^{\lambda-1}-1}.$$
(32)

Combining Lemma 2, (31) and (32) yields

$$\begin{aligned} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i+1} \right] \right] \right\|_{(\lambda,a)} \\ &\leq \left( \rho + \frac{1-\gamma}{i+1} + \frac{\gamma \left\| \mathbf{LCB} \right\| + \gamma + \left\| \mathbf{LC} \right\| \left\| \mathbf{B} \right\|}{a^{\lambda-1} - 1} \right) \qquad (33) \\ &\times \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \left( t \right) \right] \right] \right\|_{(\lambda,a)} + \alpha. \end{aligned}$$

There exists  $M \in \mathbb{Z}_+$  such that  $(1-\gamma)/(i+1) < (1-\rho)/4$  when  $i \ge M$ . Now for i > M, (28) and (33) imply that

$$\begin{aligned} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i+1} \right] \right] \right\|_{(\lambda,a)} &\leq \frac{1+\rho}{2} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \right] \right] \right\|_{(\lambda,a)} + \alpha, \\ \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i+1} \right] \right] \right\|_{(\lambda,a)} \\ &\leq \left( \frac{1+\rho}{2} \right)^{i+1-M} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{M} \right] \right] \right\|_{(\lambda,a)} \\ &+ \sum_{j=0}^{i-M} \left( \frac{1+\rho}{2} \right)^{j} \alpha \\ &\leq \left( \frac{1+\rho}{2} \right)^{i+1-M} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{M} \right] \right] \right\|_{(\lambda,a)} \\ &+ \frac{1-\left( \left( 1+\rho \right)/2 \right)^{i+1-M}}{1-\left( \left( 1+\rho \right)/2 \right)} \alpha. \end{aligned}$$
(34)

Consequently, we obtain

$$\lim_{i \to \infty} \left\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i+1} \right] \right] \right\|_{(\lambda,a)} \le \frac{2\alpha}{1-\rho}.$$
(35)

According to the relationship (14) between the input error and output error, we have

$$\mathscr{C}\left[\mathscr{A}\left[\mathbf{e}_{i}\left(t\right)\right]\right] = \mathbf{C}\sum_{\tau=0}^{t-1}\prod_{\nu=\tau+1}^{t-1}\mathbf{A}\left(\nu\right)\mathbf{B}\mathscr{C}\left[\mathscr{A}\left[\Delta\mathbf{u}_{i}\left(\tau\right)\right]\right].$$
 (36)

Similar to the proof of Lemma 2, one can prove that

$$\max_{t \in \{0,...,T\}} a^{-\lambda t} \| \mathscr{C} \left[ \mathscr{A} \left[ \mathbf{e}_{i} \left( t \right) \right] \right] \|_{2}$$

$$\leq \| \mathbf{C} \| \| \mathbf{B} \| \max_{t \in \{0,...,T\}} a^{-\lambda t} \sum_{\tau=0}^{t-1} a^{t-1-\tau} \| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \right] \right] \|_{2}$$

$$\leq \| \mathbf{C} \| \| \mathbf{B} \| \frac{\| \mathscr{C} \left[ \mathscr{A} \left[ \Delta \mathbf{u}_{i} \right] \right] \|_{(\lambda,a)}}{a^{\lambda-1} - 1}.$$
(37)

Finally, from (29), (35), and (37), we can obtain

$$\lim_{i \to \infty} \left\| \mathscr{E} \left[ \mathscr{A} \left[ \mathbf{e}_{i+1} \right] \right] \right\|_{(\lambda,a)} \le \left\| \mathbf{C} \right\| \left\| \mathbf{B} \right\| \frac{2\alpha}{1-\rho} \frac{1}{a^{\lambda-1}-1} < \epsilon$$
(38)

because lambda can be chosen arbitrarily large in (38). This completes the proof.



FIGURE 2: The max tracking error versus iteration with 5% packet dropout.

*Remark 4.* In this paper, we consider D-type iterative learning control with average operator, and the result obtained can be extended to P-type iterative learning control with average operator.

#### 4. Numerical Examples

In this simulation test, let us consider system (1) and matrices given by

$$\mathbf{A}(t) = \begin{bmatrix} 0 & 1\\ -1 - 10^{-3}t & -2 - 10^{-3}t \end{bmatrix},$$
  
$$\mathbf{B} = \begin{bmatrix} 0\\ 1 \end{bmatrix}, \qquad \mathbf{C} = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$
(39)

The random noises  $\mathbf{w}_i(t)$  and  $\mathbf{v}_i(t)$  have uniform distribution on the intervals [-0.05, 0.05] and [-0.01, 0.01], respectively. In this control problem, the desired output trajectory for  $t \in \{0, 1, ..., 50\}$  is given to be  $\mathbf{y}_d(t) = \sin(2\pi t/50)$ , and the initial states  $x_{1i}(0)$  and  $x_{2i}(0)$  have uniform distribution on the intervals [-0.01, 0.01] and [-0.02, 0.02], respectively. The fixed time interval *T* is 50. The control profile of the first iteration is  $u_1(t) = 0$ . Random packet dropout in controlleractuator channel is subject to Bernoulli distribution of expected value  $\gamma$  (1 means transmission success while 0 means transmission failure).

For expected value  $\gamma = 0.95$ , we compare our algorithm with the other 2 algorithms.

Algorithm 1 (classic ILC). The control signal is constructed as

$$u_{i+1}(t) = \gamma_{i+1}(t) \left( u_i(t) + \operatorname{Le}_i(t+1) \right) + \left( 1 - \gamma_{i+1}(t) \right) u_{i+1}(t-1)$$
(40)

with **L** = 0.2 satisfying  $|| 1 - \gamma LCB || = 0.81 < 1$ .



FIGURE 3: The mathematical expectation of the tracking error versus iteration with 5% packet dropout.

*Algorithm 2* (zero-input scheme with average ILC). The control signal is constructed as

$$u_{i+1}(t) = \gamma_{i+1}(t) \left( \mathscr{A} \left[ u_i(t) \right] + (i+1) L \mathscr{A} \left[ e_i(t+1) \right] \right)$$
(41)

with L = 0.2.

Algorithm 3. Now, we consider the proposed algorithm. From (2) and (5), the control signal  $u_{i+1}(t)$  is constructed as

$$u_{i+1}(t) = \gamma_{i+1}(t) \,\widehat{u}_{i+1}(t) + (1 - \gamma_{i+1}(t)) \,u_{i+1}(t-1)$$
  
=  $\gamma_{i+1}(t) \left( \mathscr{A} \left[ u_i(t) \right] + (i+1) \, \mathbf{L} \mathscr{A} \left[ e_i(t+1) \right] \right)$   
+  $(1 - \gamma_{i+1}(t)) \,u_{i+1}(t-1)$ , (42)

where the learning gain L = 0.2 and expected value  $\gamma = 0.95$ .

As shown in Figure 2, the tracking error profiles for the proposed algorithm are much lower than the other two algorithms with 5% packet dropout. In Figure 3, the mathematical expectation of the tracking error versus iterations is shown, and the proposed hold-input scheme with average ILC achieves the convergent performance.

# 5. Conclusion

In this work we address a remote control system problem with random packet dropout in controller-actuator channel. The hold-input scheme with average ILC is applied to handle this remote control problem with repeated tracking tasks. Through analysis we illustrate the desired convergence property of the hold-input scheme with average ILC. In our future work, we will also explore the extension to more generic stochastic process such as Markov chain.

### Nomenclature

- $\mathbb{R}$ : The set of all real numbers
- $\mathbb{Z}_+$ : The set of all positive integers
- $\mathscr{A}[\mathbf{v}_m]$ : The average operator  $\mathscr{A}[\mathbf{v}_m] = (1/m) \sum_{i=1}^m \mathbf{v}_i$
- $\mathscr{E}[\cdot]$ : The expected value of a random variable
- $\mathscr{P}[\cdot]$ : The probability of an event
- $\|\cdot\|$ : The maximal singular value of a matrix
- $\|\cdot\|_2$ : The Euclidean norm of a vector
- I: Identity matrix of appropriate dimensions
- **0**: Zero matrix of appropriate dimensions.

# Acknowledgment

This work is supported by the 973 program of China (Grant no. 2009CB320603).

# References

- S. Arimoto, S. Kawamura, and F. Miyazaki, "Berrering operation of robots by learning," *Journal of Robot Systems*, vol. 1, no. 1, pp. 123–140, 1984.
- [2] Z. Bien and K. M. Huh, "Higher-order iterative learning control algorithm," *IEE Proceedings D*, vol. 136, no. 3, pp. 105–112, 1989.
- [3] Y. Chen and K. L. Moore, "Harnessing the nonrepetitiveness in iterative learning control," in *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 3350–3355, Las Vegas, Nev, USA, December 2002.
- [4] K. L. Moore, Y. Chen, and H.-S. Ahn, "Iterative learning control: A tutorial and big picture view," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 2352–2357, December 2006.
- [5] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control: A learning-based method for highperformance tracking control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [6] Y. Chen, Z. Gong, and C. Wen, "Analysis of a high-order iterative learning control algorithm for uncertain nonlinear systems with state delays," *Automatica*, vol. 34, no. 3, pp. 345– 353, 1998.
- [7] A. Tayebi and M. B. Zaremba, "Robust iterative learning control design is straightforward for uncertain LTI systems satisfying the robust performance condition," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 101–106, 2003.
- [8] C. Yin, J.-X. Xu, and Z. Hou, "A high-order internal model based iterative learning control scheme for nonlinear systems with time-iteration-varying parameters," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2665–2670, 2010.
- [9] S. S. Saab, "A discrete-time stochastic learning control algorithm," *IEEE Transactions on Automatic Control*, vol. 46, no. 6, pp. 877–887, 2001.
- [10] M. Sun and D. Wang, "Iterative learning control with initial rectifying action," *Automatica*, vol. 38, no. 8, pp. 1177–1182, 2002.
- [11] Y. Chen, C. Wen, Z. Gong, and M. Sun, "An iterative learning controller with initial state learning," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 371–376, 1999.
- [12] R. Chi, Z. Hou, and J. Xu, "Adaptive ILC for a class of discretetime systems with iteration-varying trajectory and random initial condition," *Automatica*, vol. 44, no. 8, pp. 2207–2213, 2008.

- [13] J.-X. Xu and Y. Tan, *Linear and Nonlinear Iterative Learning Control*, vol. 291 of *Lecture Notes in Control and Information Sciences*, Springer, Berlin, Germany, 2003.
- [14] H.-S. Ahn, K. L. Moore, and Y. Chen, Iterative Learning Control: Robustness and Monotonic Convergence for Interval Systems, Communications and Control Engineering Series, Springer, London, UK, 2007.
- [15] L.-S. Hu, T. Bai, P. Shi, and Z. Wu, "Sampled-data control of networked linear control systems," *Automatica*, vol. 43, no. 5, pp. 903–911, 2007.
- [16] M. B. G. Cloosterman, N. van de Wouw, W. P. M. H. Heemels, and H. Nijmeijer, "Stability of networked control systems with uncertain time-varying delays," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1575–1580, 2009.
- [17] J. Nilsson, Real time control systems with delay [Ph.D. thesis], Lund Institute of Technology, 1998.
- [18] L. A. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562–1572, 2004.
- [19] D. Wu, J. Wu, and S. Chen, "Robust H<sub>∞</sub> control for networked control systems with uncertainties and multiple-packet transmission," *IET Control Theory & Applications*, vol. 4, no. 5, pp. 701–709, 2010.
- [20] L. Bakule and M. De La Sen, "Decentralized stabilization of networked complex composite systems with nonlinear perturbations," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA '09)*, pp. 2272–2277, Christchurch, New Zealand, December 2009.
- [21] L. Bakule and M. De La Sen, "Decentralized resilient  $H_{\infty}$  observer-based control for a class of uncertain interconnected networked systems," in *Proceedings of the American Control Conference (ACC '10)*, pp. 1338–1343, Baltimore, Md, USA, July 2010.
- [22] H.-S. Ahn, K. L. Moore, and Y. Chen, "Discrete-time intermittent iterative learning controller with independent data dropouts," in *Proceedings of the 17th World Congress, International Federation of Automatic Control (IFAC '08)*, Seoul, Korea, July 2008.
- [23] H.-S. Ahn, K. L. Moore, and Y. Chen, "Stability of discretetime iterative learning control with random data dropouts and delayed controlled signals in networked control systems," in *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV '08)*, pp. 757–762, December 2008.
- [24] C. Liu, J. Xu, and J. Wu, "Iterative learning control for network systems with communication delay or data dropout," in *Proceedings of the 48th IEEE Conference on Decision and Control Held Jointly with the 28th Chinese Control Conference (CDC/ CCC '09)*, pp. 4858–4863, Shanghai, China, December 2009.
- [25] C. Liu, J. Xu, and J. Wu, "Iterative learning control for remote control systems with communication delay and data dropout," *Mathematical Problems in Engineering*, vol. 2012, Article ID 705474, 14 pages, 2012.
- [26] X. Bu, Z. Hou, and F. Yu, "Stability of first and high order iterative learning control with data dropouts," *International Journal* of Control, Automation and Systems, vol. 9, no. 5, pp. 843–849, 2011.
- [27] Y.-J. Pan, H. J. Marquez, T. Chen, and L. Sheng, "Effects of network communications on a class of learning controlled nonlinear systems," *International Journal of Systems Science*, vol. 40, no. 7, pp. 757–767, 2009.

# Journal of Applied Mathematics

- [28] X. Bu, F. Yu, Z. Hou, and F. Wang, "Iterative learning control for a class of nonlinear systems with random packet losses," *Nonlinear Analysis: Real World Applications*, vol. 14, no. 1, pp. 567–580, 2013.
- [29] L.-X. Huang and Y. Fang, "Convergence analysis of wireless remote iterative learning control systems with dropout compensation," *Mathematical Problems in Engineering*, vol. 2013, Article ID 609284, 9 pages, 2013.

# Research Article

# Pareto Optimal Solutions for Stochastic Dynamic Programming Problems via Monte Carlo Simulation

# R. T. N. Cardoso,<sup>1</sup> R. H. C. Takahashi,<sup>2</sup> and F. R. B. Cruz<sup>3</sup>

<sup>1</sup> Departamento de Física e Matemática, Centro Federal de Educação Tecnológica de Minas Gerais, 30510-000 Belo Horizonte, MG, Brazil

<sup>2</sup> Departamento de Matemática, Universidade Federal de Minas Gerais, 31270-901 Belo Horizonte, MG, Brazil

<sup>3</sup> Departamento de Estatística, Universidade Federal de Minas Gerais, 31270-901 Belo Horizonte, MG, Brazil

Correspondence should be addressed to R. T. N. Cardoso; rodrigoc@des.cefetmg.br

Received 30 May 2013; Accepted 28 September 2013

Academic Editor: Lotfollah Najjar

Copyright © 2013 R. T. N. Cardoso et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A heuristic algorithm is proposed for a class of stochastic discrete-time continuous-variable dynamic programming problems submitted to non-Gaussian disturbances. Instead of using the expected values of the objective function, the randomness nature of the decision variables is kept along the process, while Pareto fronts weighted by all quantiles of the objective function are determined. Thus, decision makers are able to choose any quantile they wish. This new idea is carried out by using Monte Carlo simulations embedded in an approximate algorithm proposed to deterministic dynamic programming problems. The new method is tested in instances of the classical inventory control problem. The results obtained attest for the efficiency and efficacy of the algorithm in solving these important stochastic optimization problems.

# 1. Introduction

Since Bellman [1] proposed the dynamic programming technique for multistage optimization problems, his method has been very successfully applied to a wide range of problems, as it is well documented (see, for instance, [1–5]). This paper considers a dynamic programming-based method to solve stochastic discrete-time continuous-variable dynamic programming problems. A standard formulation of this problem is given by

$$\min_{u[0],\dots,u[n-1]} E\left[\sum_{k=0}^{n-1} g_k\left(x\left[k\right], u\left[k\right]\right) + g_n\left(x\left[n\right]\right)\right], \quad (1)$$

subject to

$$x [k+1] = f (x [k], u [k], w [k]), \quad k = 0, 1, ..., n-1,$$
$$x [0] = x_0, \qquad x [n] = x^*,$$
(2)

in which k = 0, 1, ..., n are the time stages, x[0], ..., x[n] are the state variables, u[0], ..., x[n - 1] are the decision

variables,  $w[0], \ldots, w[n - 1]$  are the disturbance variables, f(x[k], u[k], w[k]) is the state equation,  $g_k(x[k], u[k])$  is the cost associated to stage k,  $g_n(x[n])$  is the cost associated to the final stage, n, and E[X] is the expectation of random variable X.

The state variables track the system dynamics throughout the states, and the decision variables are actions that should be taken in order to achieve the optimization objective. Additionally, the disturbance variables are assumed as random variables, from any given distribution. The objective function is assumed separable in all variables and stages.

In discrete-variable dynamic programming problems, a resolution algorithm might involve a sequential search in a graph, the nodes being the admissible states, the arcs being the possible control actions, and the arc costs being the respective transition probabilities. Although accurate, such an algorithm certainly would have a tremendous computational cost, suffering from what is known as the "curse of dimensionality." That is, the processing costs would increase exponentially with the number of possible states (for details, see [1, 2, 6]).

In continuous-variable dynamic programming problems, an exact solution can be obtained only when the system dynamics is linear and the objective function is quadratic. In the general case, only approximate solutions are possible, which are usually based on a state space discretization, such as a uniform grid discretization of points over the state space. However, this is also not an efficient procedure due to the exponential growth of the number of states. In order to cope with this problem, computationally efficient approximate discretization procedures have been tried out already (see, for instance, [7, 8]).

In this paper, we propose numerical heuristic solutions, coupled with Monte Carlo simulations, obtainable within a quite reasonable amount of computational effort. Monte Carlo simulation is a well-known and useful method to determine probabilities by using highly intensive computational experiments [9]. Solving a dynamic programming problem by simulation is not a novelty. However, while widely used in many contexts, Monte Carlo simulations have not been used often in solving dynamic programming problems, perhaps because of their high computational costs. Just recently, now coupled with heuristics and approximations, Monte Carlo simulations started to be considered again. Its prohibitive computational costs were exchanged by solutions without strict guarantee of optimality.

Indeed, neurodynamic programming is a well-known dynamic programming approach that employs Monte Carlo sampling in stochastic settings [10]. Another very successful example is reported by de Farias and van Roy [11], which reformulated the stochastic dynamic programming problem as a linear programming problem and approximated the large resulting number of constraints by Monte Carlo sampling. Also Thompson and Cluett [12] considered Monte Carlo simulation to calculate integrals related to the expected value of the objective function of a unidimensional dualoptimal control problem that has to be decided by iterative dynamic programming [13]. A number-theoretic method, called quasi-Monte Carlo, uses number theory and numerical analysis to generate a point set that is uniformly spaced. This technique has been successfully used by Cervellera et al. [7, 8], along with neural networks, to improve a computationally tractable approximate method of discretization. Those are few examples of well-succeeded reductions in the "curse of dimensionality."

For most of the stochastic dynamic optimization problems considered in the literature, a usual model for disturbances is a zero mean and  $\sigma^2$  variance Gaussian white noise, although in some cases a nonzero mean may also appear [14]. These assumptions can be suitable for some realworld applications but it certainly will not be the case for many of them. The focus of this paper is stochastic discretetime continuous-variable dynamic programming problems submitted to non-Gaussian probability distribution functions for disturbances w[k]. In those problems, the analytical form of the expected value of the objective function may be really complex or even lead to solutions impossible to be found by means of classic dynamic programming techniques. If disturbances w[k] are Gaussian, it is possible that the best optimization reference is the expected value of the objective function. However, in many practical applications, mainly under non-Gaussian disturbances, it is almost certain that other quantiles, greater or smaller than the median, are better, as we will show shortly. Thus, a multiobjective approach seems to be quite justified [15]. That is, instead of finding the control sequence that optimizes the expected value of the objective function, this paper proposes finding Pareto fronts instead, weighted by all quantiles of the objective function.

In multiobjective optimization problems [16], there may not be a single solution that is the best (or the global optimum) with respect to all the objectives. The presence of multiple objectives in a problem usually gives rise to a family of nondominated solutions, called Pareto optimal set, in which each objective can only be improved by degrading at least one of the other objectives. Since none of the solutions in the nondominated set is absolutely better than the others, any one of them is an acceptable solution. For instance, Li and Haimes [17] presented a survey on multiobjective dynamic programming, and Trzaskalik and Sitarz [18] proposed a procedure that considers a partially ordered criteria structure in dynamic programming. However, we remind that the approach proposed here is out of the traditional multiobjective discrete-time dynamic programming methods.

A highlight of the contributions of this paper is:

- (i) We propose a scheme based on Monte Carlo simulations coupled with a deterministic discrete-time dynamic programming method, which is exact for continuous-variable problems and asymptotically accurate for discrete-variable problems. Details will be given shortly, but the main idea of this dynamic programming method is to study the system dynamics as an iteration operated on closed sets, essentially considering the problem from a geometrical point of view, instead of using a more traditional way of studying it in the arcs of the graph. Since the scheme proposed is based on simulations, it will be possible to use non-Gaussian probability distribution functions for disturbances.
- (ii) Using the proposed scheme, a multiobjective approach is employed, because why should one ever consider only the expected value as the reference for the optimization when it is possible to take into account all quantiles? To deal with this point, a Pareto front is presented as the answer of the problem, weighted by a function of the empirical quantiles of the decision variables. This is true because quantiles are functions of the random variables, which can be sampled by Monte Carlo simulations.

The proposed methodology and the algorithm are explained in detail in Section 2. A case study based on a classical inventory control problem is conducted and presented in Section 3. Finally, in Section 4, we summarize this paper and conclude it with topics for future research in the area.

algorithm
read input;
for $i = 1$ until sampleSize do
generate randomly a sequence of disturbances with a
given probability distribution function;
find a sequence of decision variables that optimizes
the objective-function, as if it were a
deterministic dynamic programming problem;
end for
mount the Pareto front of the decision variables, weighted by its quantiles;
take the box-plot, the average, or any other quantile of
these variables as the answer of the problem.
end algorithm

Algorithm 1

### 2. Proposed Approach

We propose here a computationally tractable scheme for stochastic dynamic programming problems formulated as in (1). Our heuristic method consists of a multiobjective approach based on Monte Carlo simulations embedded in a deterministic dynamic programming method. For our convenience, the deterministic method chosen here was a geometrical algorithm, described in detail in Cardoso [19] and Cardoso et al. [15], which is exact for continuousvariable problems and asymptotically exact for discretevariable problems.

In the geometrical algorithm described by Cardoso [19], the system dynamics is supposed to be linear, with real variables, or else with integer variables that could be relaxed. The objective functions could be of any type, not necessarily quadratic, although quadratics are the most used functions in the literature. The geometrical algorithm is inspired by approximate dynamic programming methods, namely, the certainty equivalent control technique and the model predictive control technique (details may be found in [2]). An openloop optimal control computation is used in conjunction with a rolling horizon, which means that more than one control move is generally calculated. Finally, we remark that the scheme is flexible enough to work under any other fast deterministic dynamic programming method as well. The proposed algorithm may be summarized as shown in Algorithm 1.

We remind that because the proposed procedure is based on simulations, it would be no trouble to consider disturbances from any probability distribution, including those for which an analytical solution is difficult, impossible, or mathematically intractable. For example, it is possible to consider multivariate-Gaussian distributions with different means and variances per coordinate, or even multimodal and asymmetrical distributions. This fact is very convenient in accurately representing real-life phenomena. The multiobjective approach explicitly considers the random distribution of the control variables and implicitly takes into account the random distribution of the state variables and of the objective functions. In other words, all variables are to be treated as random, which is possible because they are functions of the disturbance random variables. As a result, we can find boxplots, histograms, and quantiles for all decision variables, as outcomes of the proposed method for the problem at hand.

As a final remark, according to the formulation presented in (1), Monte Carlo simulations should have been used to compute the sequence of control variables to optimize the expected value of the objective function for several disturbances randomly generated. Instead, the proposed algorithm presents the expected value of a sequence of decision variables to optimize the objective function, for several randomly generated disturbances. In the appendix, we present a theorem that shows that (i) the result found by the proposed methodology is a lower bound for a classical solution method and (ii) the equality is valid if the decision variable minimizes the objective function for almost all disturbances. Under a practical point of view, the equality occurs, for example, if the decision variables are constrained to compact sets, as usually happens in linear problems that are solved by numerical methods. Finally, we remind that the convergence of this algorithm relies on the convergence of Monte Carlo simulations [9].

#### 3. Simulations

3.1. Preliminaries. The algorithm was implemented in MAT-LAB (MATLAB is a trademark of The MathWorks, Inc.), and all cases were run in a common PC. For all simulations, 5,000 disturbance vectors w[k] were generated. However, with only 100 vectors w[k], convergence could be verified [20]. The purpose of these case studies is just to show a simple and yet interesting application of the proposed methodology. Then, a classical stochastic dynamic programming example is considered, namely, the inventory control. Details may be found in Bertsekas [2], but in few words the problem consists in determining optimal orders to be placed for some items at discrete-time stages so as to meet a stochastic demand. Moreover, it is requested that the available stock at the final stage is null. For such a problem, the variables are inherently discrete since items are counted, but the range of levels for



FIGURE 1: Case 1, box-plots for first product optimal orders.

an item is too large to be practical for a discrete-variable dynamic programming solution. Then, the discrete variables will be relaxed and they will be considered as real numbers. The initial stock will also be a decision variable. This dynamic optimization problem can be formulated as

$$\min_{u[0],\dots,u[n-1]} E\left[\sum_{k=0}^{n-1} \left(c_k x \left[k\right] + d_k u \left[k\right]\right) + c_n x \left[n\right]\right], \quad (3)$$

subject to

$$x [k+1] = x [k] + u [k] - w [k], \quad k = 0, 1, ..., n-1,$$
$$x [k] \ge 0, \qquad u [k] \ge 0, \qquad (4)$$
$$x [0] = x_0, \qquad x [n] = x^*,$$

in which each stage k corresponds to each month, n is the horizon, each state vector x[k] represents the inventory level (the stock available) at the beginning of stage k, each control action vector u[k] is the amount ordered at the beginning of stage k, and each disturbance vector w[k] is a stochastic customer demand during stage k, given from some probability distribution, each vector having size mcorresponding to the number of products considered.

The inventory level evolves according to the linear discrete-time dynamic system:

$$x[k+1] = x[k] + u[k] - w[k].$$
(5)

A linear cost function per stage is composed by a penalty for each positive stock, represented by a row per unit cost vector  $c_k$ , added to the purchasing cost, represented by a row per unit cost vector  $d_k$ .

We consider the following three instances of the stochastic inventory control problem, all of them having an optimization horizon n = 12, vectors  $c_k$  and  $d_k$  unitary, and programming goal  $x^*$  equal to the null vector. The decision criterion of the multiobjective approach is the trade-off



FIGURE 2: Case 1, band of values for first product optimal orders.

between the probabilities of having some stock at the end and the probability of not attending some demand. The instances are as follows.

*Case 1.* Customer demands follow a Gaussian probability distribution, with multiproducts and a coupling constraint. Case 1 is composed by m = 3 products. Demands follow Gaussian distributions, with mean 30 and standard deviation 10, for first product, mean 60 and standard deviation 10, for second, and mean 30 and standard deviation 20, for third. The coupling constraint says that 2 units of product 1 and 1 unit of product 2 are necessary to produce each unit of product 3.

*Case 2.* Customer demands follow a bimodal probability distribution function. Case 2 is composed by just n = 1 product, with demands following a mix of Gaussian distributions having mean 100 and standard deviation 10, with probability 0.3, and mean 200 and same standard deviation, with probability 0.7.

*Case 3.* Customer demands follow an asymmetric probability distribution function. Also, only n = 1 product is considered, with demand that follows a zero-truncated log-normal distribution, with mean 2 and standard deviation 1.

*3.2. Results for Case 1.* For Case 1, described earlier, the simulation results are presented in Figures 1–5. Only results for the first product are shown, as similar outputs (not presented) were obtained for the other two products. Figure 1 shows box-plots for the first product optimal order, and Figure 2 presents the corresponding band of values, between quantiles 0.3 and 0.7. As expected, it is noticeable here an increase in the uncertainty of the orders, what is evident from



FIGURE 3: Case 1, box-plots for first product inventory levels when ordering by the median.



FIGURE 4: Case 1, box-plots for first product inventory levels when ordering by 70th percentile.

the band enlargement that is observed as time goes by. Figures 3 and 4 illustrate what is expected to happen to the inventory level, when the orders are made by the median values and by quantile 0.7, respectively. These are values easily obtainable from the algorithm. From Figure 3, we can roughly estimate that 50%-50% are the probabilities of having and not having stock by the end of the planning horizon, which is expected because of the probability distribution of the demands being symmetrical. We see here a reduction in the probability of a negative stock, in case product orders would be placed by quantile 0.7, Figure 4. Actually, the decision maker could play with these two conflicting objectives, namely, (i) a low probability of having stock at the end of the horizon period and (ii) a low probability of not having stock at all, which would mean that some demand would not be attended. Figure 5 presents the respective Pareto front that resulted from our simulations.



FIGURE 5: Case 1, Pareto front for a trade-off between the probabilities of having and not having stock.



FIGURE 6: Case 2, box-plots of optimal orders.

3.3. Results for Case 2. Results for Case 2 are depicted in Figures 6–10. The box-plots for optimal orders and respective band of values are presented in Figures 6 and 7, which follow basically the same pattern observed for Case 1; that is, we observe an increase in the width (uncertainty) of the band of values to be ordered along the time. Comparing two strategies of placing orders, that is, from the median values or else from percentile 0.7, leads to the results presented in Figures 8 and 9, from which we can notice that, by the final stage, the probabilities of having and not having stock are not 50%-50% anymore, for orders taken from median values. This results from the asymmetry of demands. On the other hand, ordering from quantile 0.7 will increase significantly the probability of having stock at the final stage. Of course, we could also estimate these probabilities for any ordering strategy we wish, as seen in Figure 10. As a final remark, it can be seen in this case that bimodality in demands creates



FIGURE 7: Case 2, band of values for optimal orders.



FIGURE 8: Case 2, box-plots for inventory levels when ordering by the median.

some difficulty for traditional approaches for planning order placements, as a median-based order no longer guarantees 50%-50% probabilities of having and not having stock by the end of the stages.

3.4. Results for Case 3. Figures 11, 12, 13, 14, and 15 show the results for Case 3, which accounts for asymmetrical distributed demands. In this case, orders taken by the median lead to approximately 82% of probability of having stock at the final stage and, of course, 18% of probability of not having stock at all at the final stage. On the other hand, orders taken by quantile 0.7 lead to approximately 37% of probability of having stock at the end of stages and 63% of not having any stock at the end. By means of our methodology, it would



FIGURE 9: Case 2, box-plots for inventory levels when ordering by 70th percentile.



FIGURE 10: Case 2, Pareto front for a trade-off between the probabilities of having and not having stock.

be easy to identify, for instance, that the target of 50%-50% would be reached if orders were to be placed from quantile  $\approx$  0.64.

#### 4. Discussions and Conclusions

In this paper, we proposed a multiobjective approach for stochastic discrete-time real-variable dynamic programming problems, which is based on Monte Carlo simulations coupled with a deterministic dynamic programming algorithm. Our approach was shown to be able to deliver suboptimal (heuristic) solutions by using common desktop computers, within a reasonable amount of computational time. Once a Pareto front for the problem becomes built, decision makers can choose any quantile that is perceived to be advantageous for the specific situation. In addition, a band of values,



FIGURE 11: Case 3, box-plots of the optimal orders.



FIGURE 12: Case 3, band of values for optimal orders.

histograms of the decision variables, and their box-plots are provided as outcomes of the proposed method for a given problem, instead of just the decision variable values that optimize the expected value of the objective function. For a more effective understanding of the method proposed and the results delivered, we presented a case study based on a classical inventory control problem. The numerical results obtained in the simulations showed that Monte Carlo simulations were quite effective in solving realistic cases and that the methodology is a promising technique.

Future research will investigate the application of the proposed methodology in instances of real-world dynamic problems larger than those ones treated here. Additionally, more realistic and complex modeling of the random variables



FIGURE 13: Case 3, box-plots for inventory levels when ordering by the median.



FIGURE 14: Case 3, box-plots for inventory levels when ordering by 70th percentile.



FIGURE 15: Case 3, Pareto front for trade-off between the probabilities of having and not having stock.

will be considered, for instance, with variable-dependency. These are some few topics for future research in this area.

#### Appendix

This Appendix presents a theorem that validates the proposed algorithm. The random variables are considered to be continuous but the discrete variable case could be treated similarly. The improper integral, which comes from the expected value of the continuous random variable, is supposed to converge. If a minimum does not exist, just replace it by the infimum. All minimization could be for almost all w instead of for all w. The theorem is based on a review of the Fatou Lemma, as follows.

**Lemma A.1** (review of Fatou Lemma). Let  $f : \mathbb{R}^2 \to \mathbb{R}$  be an integrable function in the first coordinate and with minimum in the second one. Thus,

$$\int \min_{u} f(w, u) \, dw \le \min_{u} \int f(w, u) \, dw.$$
(A.1)

Proof. For each fixed w, consider

$$\overline{u}(w) = \arg\min_{u} f(w, u).$$
(A.2)

Then, one has that

$$f(w,\overline{u}(w)) \le f(w,u), \quad \forall u.$$
 (A.3)

As the integral operator preserves monotonicity

$$\int \min_{u} f(w, u) dw = \int f(w, \overline{u}(w)) dw$$

$$\leq \int f(w, u) dw, \quad \forall u.$$
(A.4)

As this expression holds for all *u*, it holds for

$$u_o = \arg\min_u \int f(w, u) \, dw. \tag{A.5}$$

Thus,

$$\int \min_{u} f(w, u) \, dw \le \min_{u} \int f(w, u) \, dw. \tag{A.6}$$

In other words, the expected value of the minimum is always a lower bound for the minimum of the expected value.

**Theorem A.2** (validation of the algorithm). Let  $f : \mathbb{R}^2 \to \mathbb{R}$  be an integrable function in the first coordinate and with minimum in the second one. If there exists  $\overline{u}$  that does not depend on w, such that

$$f(w,\overline{u}) = \min_{u} f(w,u), \quad \forall w, \tag{A.7}$$

the following equality holds:

$$\min_{u} f(w,u) dw = \min_{u} \int f(w,u) dw.$$
(A.8)

*Proof.*  $(\Rightarrow)$  Suppose that there exists

$$\overline{u} = \arg\min_{u} f(w, u), \quad \forall w.$$
(A.9)

Consider

$$u_o = \arg\min_u \int f(w, u) \, dw. \tag{A.10}$$

By Lemma A.1,

$$\int f(w,\overline{u}) dx = \int \min_{u} f(w,u) dw$$
  
$$\leq \min_{u} \int f(w,u) dw = \int f(w,u_{o}) dw.$$
(A.11)

As  $u_0$  minimizes the integral, it is true that

$$f(w, u_o) dw \le \int f(w, u) dw, \quad \forall u.$$
(A.12)

In particular, for  $\overline{u}$ , which is fixed and minimizes f, for all w

$$\int f(w, u_o) dw \le \int f(w, \overline{u}) dw.$$
(A.13)

Therefore, the following equality holds:

$$\int f(w, u_o) dw = \int f(w, \overline{u}) dw.$$
(A.14)

In other words,

$$\int \min_{u} f(w, u) \, dw = \min_{u} \int f(w, u) \, dw. \tag{A.15}$$

Thus, if minimums exist, they must coincide

$$u_o = \arg\min_{u} \int f(w, u) \, dw = \overline{u} = \arg\min_{u} f(w, u) \,. \quad (A.16)$$

(⇐) Suppose that, for all  $u_o$  given, there exists a set  $\overline{W}$  with positive measure, for which

$$u_{o} \neq \arg\min_{u} f(\overline{w}, u), \quad \forall \overline{w} \in \overline{W} \Longrightarrow$$

$$u_{o} \neq \arg\min_{u} f(w, u), \quad \forall w.$$
(A.17)

In other words,  $\forall \overline{w} \in \overline{W}$ , there exists a  $\overline{u}$  which depends on  $\overline{w}$ , such that

$$\overline{u}(\overline{w}) = \arg\min_{u} f(\overline{w}, u).$$
(A.18)

Therefore,

$$f(\overline{w}, \overline{u}(\overline{w})) < f(\overline{w}, u_o), \quad \forall \overline{w} \in \overline{W}.$$
(A.19)

By the monotonicity of the integral

$$\int_{\overline{W}} f(w,\overline{u}) dw = \int_{\overline{W}} \min_{u} f(w,u) dw$$
$$< \int_{\overline{W}} f(w,u_{o}) dw, \quad \forall u_{o} \Longrightarrow \quad (A.20)$$
$$\min_{u} f(w,u) dw < \int f(w,u_{o}) dw, \quad \forall u_{o}.$$
In particular,

$$u_o = \arg\min_u \int f(w, u) \, dw. \tag{A.21}$$

Thus, the inequality just holds:

$$\int \min_{u} f(w, u) \, dw < \min_{u} \int f(w, u) \, dw.$$
(A.22)

Therefore, the equality is valid, for example, if f is linear and u is considered constrained to compact sets.

### Acknowledgments

The authors would like to thank the Brazilian agencies, CAPES, CNPq, and FAPEMIG for the financial support.

### References

- R. Bellman, *Dynamic Programming*, Dover Books on Computer Science Series, Dover, Mineola, NY, USA, 2003.
- [2] D. P. Bertsekas, Dynamic Programming and Optimal Control, vol. 3, Athena Scientific, Belmont, Mass, USA, 3rd edition, 2007.
- [3] B. H. Dias, A. L. M. Marcato, R. C. Souza et al., "Stochastic dynamic programming applied to hydrothermal power systems operation planning based on the convex hull algorithm," *Mathematical Problems in Engineering*, vol. 2010, Article ID 390940, 20 pages, 2010.
- [4] Y. Zhang, S. Song, C. Wu, and W. Yin, "Dynamic programming and heuristic for stochastic uncapacitated lot-sizing problems with incremental quantity discount," *Mathematical Problems in Engineering*, vol. 2012, Article ID 582323, 21 pages, 2012.
- [5] S. Ji, C. Sun, and Q. Wei, "The dynamic programming method of stochastic differential game for functional forward-backward stochastic system," *Mathematical Problems in Engineering*, vol. 2013, Article ID 958920, 14 pages, 2013.
- [6] D. P. Bertsekas, Dynamic Programming: Deterministic and Stochastic Models, Prentice Hall, New York, NY, USA, 1987.
- [7] C. Cervellera, V. C. P. Chen, and A. Wen, "Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient state space discretization," *European Journal of Operational Research*, vol. 171, no. 3, pp. 1139–1151, 2006.
- [8] C. Cervellera, A. Wen, and V. C. P. Chen, "Neural network and regression spline value function approximations for stochastic dynamic programming," *Computers & Operations Research*, vol. 34, no. 1, pp. 70–90, 2007.
- [9] S. M. Ross, *Simulation*, Academic Press, San Diego, Calif, USA, 5th edition, 2012.
- [10] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, Mass, USA, 1996.
- [11] D. P. de Farias and B. van Roy, "Approximate linear programming for average-cost dynamic programming," in *Advances in Neural Information Processing Systems 15*, S. T. Becker and K. Obermayer, Eds., pp. 1587–1594, The MIT Press, Cambridge, Mass, USA, 2003.
- [12] A. M. Thompson and W. R. Cluett, "Stochastic iterative dynamic programming: a Monte Carlo approach to dual control," *Automatica*, vol. 41, no. 5, pp. 767–778, 2005.

- [13] R. Luus, Iterative Dynamic Programming, Chapman & Hall/ CRC, New York, NY, USA, 2000.
- [14] H. Kushner, *Introduction to Stochastic Control*, Holt, Rinehart and Winston, 1971.
- [15] R. T. N. Cardoso, R. H. C. Takahashi, F. R. B. Cruz, and C. M. Fonseca, "A multi-quantile approach for openloop stochastic dynamic programming problems," in *Proceedings* of the IFAC Workshop on Control Applications of Optimization (CAO '09), vol. 7, pp. 52–57, 2009, http://www.scopus. com/inward/record.url?eid=2-s2.0-79960935120&partnerID= 40&md5=7c3ed430fc2a0c425d0c182ac0ecf0d8.
- [16] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making: Theory and Methodology*, Dover, Mineola, NY, USA, 2008.
- [17] D. Li and Y. Y. Haimes, "Multiobjective dynamic programming. The state of the art," *Control, Theory and Advanced Technology*, vol. 5, no. 4, pp. 471–483, 1989.
- [18] T. Trzaskalik and S. Sitarz, "Discrete dynamic programming with outcomes in random variable structures," *European Journal* of Operational Research, vol. 177, no. 3, pp. 1535–1548, 2007.
- [19] R. T. N. Cardoso, Algorithms for dynamic programming based on invariant families [M.S. thesis], Departamento de Matemática, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2005, (Portuguese), http://www.mat.ufmg.br/intranetatual/pgmat/TesesDissertacoes/uploaded/Diss109.pdf.
- [20] S. Robinson, "A statistical process control approach to selecting a warm-up period for a discrete-event simulation," *European Journal of Operational Research*, vol. 176, no. 1, pp. 332–346, 2007.

## **Research** Article

## **Dynamic Mean-Variance Model with Borrowing Constraint under the Constant Elasticity of Variance Process**

### Hao Chang<sup>1</sup> and Xi-min Rong<sup>2</sup>

<sup>1</sup> Department of Mathematics, Tianjin Polytechnic University, Tianjin 300387, China <sup>2</sup> School of Science, Tianjin University, Tianjin 300072, China

Correspondence should be addressed to Hao Chang; ch8683897@126.com

Received 24 April 2013; Revised 17 September 2013; Accepted 1 October 2013

Academic Editor: Rung Ching Chen

Copyright © 2013 H. Chang and X.-m. Rong. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies a continuous-time dynamic mean-variance portfolio selection problem with the constraint of a higher borrowing rate, in which stock price is governed by a constant elasticity of variance (CEV) process. Firstly, we apply Lagrange duality theorem to change an original mean-variance problem into an equivalent optimization one. Secondly, we use dynamic programming principle to get the Hamilton-Jacobi-Bellman (HJB) equation for the value function, which is a more sophisticated nonlinear second-order partial differential equation. Furthermore, we use Legendre transform and dual theory to transform the HJB equation into its dual one. Finally, the closed-form solutions to the optimal investment strategy and efficient frontier are derived by applying variable change technique.

### 1. Introduction

The main purpose of this paper is to focus on the portfolio selection problem under the constant elasticity of variance (CEV) model. The CEV model was originally proposed by Cox and Ross [1] as an alternative diffusion process for European option pricing. It is a natural extension of the geometric Brownian motion (GBM). The advantage of the CEV model is that it can explain the empirical bias exhibited by Black and Scholes [2] model, such as volatility smile. Therefore, the CEV model was often applied to analyze the option pricing formulas, for example, Schroder [3], Phelim and Yisong [4], Davydov and Linetsky [5], and so forth. Recently, the CEV model has been introduced into annuity contracts to study the optimal investment strategy in the defined contribution and defined benefit pension plan (referring to Xiao et al. [6], Gao [7, 8]), but those models were all considered in the utility function framework. In the existing literatures, as far as our knowledge, the CEV model in the mean-variance framework has not been reported.

In most of the real-world situations, different interest rates for borrowing and lending are often faced by the investors. It is clear that portfolio selection models with borrowing constraint will make it more practical. This attracts the attention of many authors, referring to Paxson [9], Fleming and Zariphopoulou [10], Vila and Zariphopoulou [11], Teplá [12], and Zariphopoulou [13]. However, those models were usually dealt with under expect utility criterion, and the risky asset price was usually supposed to be driven by a GBM. In addition, the risk and return relationship is implicit in the utility function approach and cannot be disentangled at the different level of the optimal strategy. As a matter of fact, the optimal investment strategy under the utility maximizing criterion is not necessarily mean-variance efficient.

This paper introduces a CEV model and borrowing constraint into the classical portfolio selection problem in a continuous-time mean-variance framework. For the mean-variance portfolio selection problem, stochastic linearquadratic (LQ) control technique is an effective method (e.g., Zhou and Li [14], Li et al. [15], and Chiu and Li [16], Xie et al. [17]). But borrowing constraint forces this problem to become piecewise linear-quadratic and is hence no longer a LQ control problem (see [18]). In addition, the introduction of the CEV model gives rise to some new difficulties, which are not easily dealt with in solving the associated HJB equation. In this paper, we firstly apply Lagrange duality theorem to change an original mean-variance problem into an equivalent optimization one. Secondly, we use dynamic programming principle to get the Hamilton-Jacobi-Bellman (HJB) equation for the value function, which is a more sophisticated nonlinear second-order partial differential equation. Further, we use Legendre transform and dual theory to transform the HJB equation into its dual one. Finally, the closed-form solutions to the optimal investment strategy and efficient frontier are derived by applying variable change technique. There are several innovations in this paper: (i) stock price is supposed to follow the CEV model, which is a natural extension of geometric Brownian motion; (ii) we consider a dynamic mean-variance portfolio selection problem with borrowing constraints under a CEV model and assume that the borrowing rate is larger than the risk-free interest rate; (iii) the closed-form solutions to the optimal investment strategy and the efficient frontier are obtained.

The paper is organized as follows. In Section 2, we introduce a CEV model and describe portfolio selection problem with borrowing constraint in a mean-variance framework. In Section 3, the closed-form solution to optimal investment strategy is derived by applying Legendre transform and dual theory. Section 4 gives the main results on the optimal strategy and the efficient frontier. Section 5 concludes this paper.

### 2. Mean-Variance Model

In this paper, we consider a financial market where two assets are traded continuously over [0, T]. One asset is a bond with price  $P_t$  at time t, whose price process  $P_t$  with borrowing constraint can be expressed in the following ordinary differential equation (see Fu et al. [18]):

$$dP_t = \begin{cases} rP_t dt, & \text{if } P_t \ge 0, \ t \in [0,T] \\ RP_t dt, & \text{if } P_t < 0, \ t \in [0,T] \end{cases}, \quad P_0 = p_0 > 0, \quad (1)$$

where the constant r > 0 is the interest rate of the bond and R is the borrowing rate being larger than r.

Letting  $x^{-} = -\min(x, 0)$ , then (1) can be rewritten as

$$dP_t = (rP_t - (R - r)P_t^{-})dt, \quad P_0 = p_0 > 0.$$
 (2)

The another asset is a stock with prices  $S_t$  at time t, whose price process  $S_t$  is supposed to follow the constant elasticity of variance (CEV) model:

$$dS_{t} = S_{t} \left[ \mu dt + k S_{t}^{\beta} dW_{t} \right], \quad S_{0} = s_{0} > 0, \quad (3)$$

where  $\mu$  ( $\mu > R > r$ ) is the instantaneous return rate of the stock. k and  $\beta$  are constant parameters, the elasticity parameter  $\beta$  satisfies the general condition:  $\beta \leq 0$ .  $kS_t^{\beta}$ is defined as the instantaneous volatility of the stock, and  $W_t$  is a one-dimensional standard and adapted Brownian motion defined on the filtered complete probability space  $(\Omega, \mathcal{F}, P, \{\mathcal{F}_t\}_{t>0})$ .

*Remark 1.* Note that there are four special interpretations for the elasticity parameter  $\beta$ :

- (i) if β = 0, the CEV model is reduced to a geometric Brownian motion (GBM);
- (ii) if  $\beta = -1$ , it is the Ornstein-Uhlenbeck process;
- (iii) if  $\beta = -1/2$ , it is the model first presented by Cox and Ross [1] as an alternative diffusion process for valuation of options;
- (iv) if  $\beta < 0$ , this means that the instantaneous volatility  $kS_t^{\beta}$  increases as the stock price decreases and can generate a distribution with a fatter left tail (referring to Gao [7]).

Suppose that short-selling of the stock is allowed and transaction cost and consumption are not considered. We denote by  $X_t$  the wealth of the investor at time  $t \in [0, T]$  and by  $N_i(t)$  the share of asset *i*th held by the investor at time t, i = 1, 2. Let  $\pi_t = N_2(t)S_t$  be the amount invested in the stock at time  $t, t \in [0, T]$ . Clearly, the amount invested in the bond is  $\pi_t^0 = N_1(t)P_t = X_t - \pi_t$ . The wealth process  $X_t = N_1(t)P_t + N_2(t)S_t$  corresponding to trading strategy  $\pi_t$  is subject to the following stochastic differential equation:

$$dX_{t} = N_{1}(t) dP_{t} + N_{2}(t) dS_{t}$$

$$= N_{1}(t) (rP_{t} - (R - r) P_{t}^{-}) dt$$

$$+ N_{2}(t) S_{t} (\mu dt + kS_{t}^{\beta} dW_{t})$$

$$= (rN_{1}(t) P_{t} - (R - r) N_{1}(t) P_{t}^{-}) dt$$

$$+ \pi_{t} (\mu dt + kS_{t}^{\beta} dW_{t}).$$
(4)

That is, we have

$$dX_{t} = \left( rX_{t} + (\mu - r) \pi_{t} - (R - r) (X_{t} - \pi_{t})^{-} \right) dt + \pi_{t} k S_{t}^{\beta} dW_{t}, \quad X_{0} = x_{0} > 0.$$
(5)

The investor's objective is to find an optimal portfolio  $\pi_t$ such that the expected terminal wealth satisfies  $\mathbb{E}X_T = C$ , for some constant  $C \in \mathbb{R}$ , while the risk measured by the variance of the terminal wealth

$$\operatorname{Var} X_T = \mathbb{E} [X_T - \mathbb{E} X_T]^2 = \mathbb{E} (X_T - C)^2$$
(6)

is minimized. The problem of finding out such a portfolio  $\pi_t$  is referred to as the mean-variance portfolio choice problem.

In the modern portfolio selection theory, a portfolio  $\pi_t$ is said to be admissible if it is integrable and  $\{\mathcal{F}_t\}_{t>0}$ -adapted, and (5) has a unique solution corresponding to  $\pi_t$ . In this case, we refer to  $(X_t, \pi_t)$  as an admissible pair. Therefore, the meanvariance problem can be formulated as a linearly constrained stochastic optimization problem:

Minimize 
$$\operatorname{Var} X_T = \mathbb{E}(X_T - C)^2$$
  
subject to  $\mathbb{E}X_T = C$  (7)  
 $(X_t, \pi_t)$  satisfies (5).

Finally, an optimal investment strategy of the above problem is called an efficient portfolio corresponding to *C*, the corresponding (*C*, Var  $X_T$ ) is called an efficient point, whereas the set of all the efficient points, when the parameter *C* runs over  $[x_0e^{rT}, +\infty)$ , is called an efficient frontier.

*Remark 2.* If  $\pi_t < 0$ , this means that the investor is shortselling the stock. If  $X_t - \pi_t < 0$ , then the investor needs borrowing the money from the bank at interest rate *R* and the amount to borrow is  $|X_t - \pi_t|$ . Otherwise, we do not borrow the money to run the portfolio.

### 3. The Optimal Portfolio

To find out an optimal investment strategy for the problem (7) corresponding to the constraint  $\mathbb{E}X_T = C$ , we introduce a Lagrange multiplier  $2\lambda \in R$  and arrive at a new objective function:

$$\widehat{J}(\pi_t, \lambda) = \mathbb{E}\left[ \left( X_T - C \right)^2 + 2\lambda \left( X_T - C \right) \right]$$
  
=  $\mathbb{E} \left( X_T - (C - \lambda) \right)^2 - \lambda^2.$  (8)

Letting  $\gamma = C - \lambda$ , we obtain the following stochastic control problem:

Minimize  $\overline{J}(\pi_t, \gamma) = \mathbb{E}(X_T - \gamma)^2 - (C - \gamma)^2$ subject to  $(X_t, \pi_t)$  satisfies (5). (9)

The link between problem (7) and (9) is provided by Lagrange duality theorem (see Fu et al. [18] and Luenberger [19]):

Minimize 
$$\operatorname{Var} X_T = \underset{\lambda \in \mathbb{R}}{\operatorname{Max}} \underset{\pi_t}{\operatorname{Min}} \widehat{f}(\pi_t, \lambda)$$
  
$$= \underset{\gamma \in \mathbb{R}}{\operatorname{Max}} \underset{\pi_t}{\operatorname{Min}} \overline{f}(\pi_t, \gamma).$$
(10)

For a fixed constant  $\gamma$ , the problem (9) is clearly equivalent to

Minimize 
$$\mathbb{E}(X_T - \gamma)^2$$
  
subject to  $(X_t, \pi_t)$  satisfies (5). (11)

We define the value function H(t, s, x) as

$$H(t,s,x) = \underset{\pi_t}{\operatorname{Min}\mathbb{E}}\left(\left(X_T - \gamma\right)^2 \mid S_t = s, X_t = x\right)$$
(12)

with boundary condition  $H(T, s, x) = (x - \gamma)^2$ .

According to dynamic programming principle, H(t, s, x) can be taken as the continuous solution to the following HJB equation:

$$H_{t} + \mu s H_{s} + \frac{1}{2} k^{2} s^{2\beta+2} H_{ss}$$
  
+ 
$$M_{\pi_{t}} \left\{ \left[ rx + (\mu - r) \pi_{t} - (R - r) (x - \pi_{t})^{-} \right] H_{x} + \frac{1}{2} (\pi_{t} k s^{\beta})^{2} H_{xx} + k^{2} s^{2\beta+1} \pi_{t} H_{xs} \right\} = 0, \qquad (13)$$

3

where  $H_t$ ,  $H_s$ ,  $H_{ss}$ ,  $H_x$ ,  $H_{xx}$ , and  $H_{xs}$  denote first-order and second-order partial derivatives with respect to time *t*, stock price  $S_t$ , and wealth process  $X_t$ .

Let us firstly describe borrowing situation. Not borrowing and investing in the bond means that  $x - \pi_t \ge 0$  and borrowing to invest in the stock means that  $x - \pi_t < 0$ . We define the nonborrowing region  $\Theta$  in the (t, x)-plane to be

$$\Theta = \left\{ (t, x) \in [0, T] \times \mathbb{R} \mid x - \pi_t \ge 0 \right\}.$$
(14)

Taking borrowing situation into consideration, we rewrite the HJB equation (13) as

$$\begin{aligned} H_{t} + \mu s H_{s} &+ \frac{1}{2} k^{2} s^{2\beta+2} H_{ss} \\ &+ \underset{\pi_{t}}{\text{Min}} \left\{ \left[ rx + (\mu - r) \,\pi_{t} \right] H_{x} + \frac{1}{2} \left( \pi_{t} k s^{\beta} \right)^{2} H_{xx} \\ &+ k^{2} s^{2\beta+1} \pi_{t} H_{xs} \right\} = 0, \quad \text{if } (t, x) \in \Theta, \end{aligned}$$

$$\begin{aligned} H_{t} + \mu s H_{s} + \frac{1}{2} k^{2} s^{2\beta+2} H_{ss} \\ &+ \underset{\pi_{t}}{\text{Min}} \left\{ \left[ Rx + (\mu - R) \,\pi_{t} \right] H_{x} + \frac{1}{2} \left( \pi_{t} k s^{\beta} \right)^{2} H_{xx} \\ &+ k^{2} s^{2\beta+1} \pi_{t} H_{xs} \right\} = 0, \quad \text{if } (t, x) \notin \Theta. \end{aligned}$$

$$(15)$$

The optimal value  $\pi_t^*$  of (15) is given by

$$\pi_{t}^{*} = \begin{cases} \frac{-(\mu - r)H_{x} - k^{2}s^{2\beta + 1}H_{xs}}{k^{2}s^{2\beta}H_{xx}}, & \text{if } (t, x) \in \Theta, \\ \frac{-(\mu - R)H_{x} - k^{2}s^{2\beta + 1}H_{xs}}{k^{2}s^{2\beta}H_{xx}}, & \text{if } (t, x) \notin \Theta. \end{cases}$$
(16)

Putting (16) into (15), we have

$$H_{t} + \mu s H_{s} + \frac{1}{2} k^{2} s^{2\beta+2} H_{ss} + rx H_{x} - \frac{1}{2k^{2} s^{2\beta} H_{xx}} \\ \times \left[ (\mu - r) H_{x} + k^{2} s^{2\beta+1} H_{xs} \right]^{2} = 0, \quad \text{if } (t, x) \in \Theta,$$

$$H_{t} + \mu s H_{s} + \frac{1}{2} k^{2} s^{2\beta+2} H_{ss} + Rx H_{x} - \frac{1}{2k^{2} s^{2\beta} H_{xx}} \\ \times \left[ (\mu - R) H_{x} + k^{2} s^{2\beta+1} H_{xs} \right]^{2} = 0, \quad \text{if } (t, x) \notin \Theta.$$

$$(17)$$

Letting  $\tau = r$ , if  $(t, x) \in \Theta$ ;  $\tau = R$ , if  $(t, x) \notin \Theta$ , we get

$$H_{t} + \mu s H_{s} + \frac{1}{2} k^{2} s^{2\beta+2} H_{ss} + \tau x H_{x} - \frac{1}{2k^{2} s^{2\beta} H_{xx}} \left[ (\mu - \tau) H_{x} + k^{2} s^{2\beta+1} H_{xs} \right]^{2} = 0.$$
(18)

According to the convexity of the value function, we can define a Legendre transform:

$$\widehat{H}(t,s,z) = \sup_{x>0} \left\{ H(t,s,x) - zx \right\},$$
(19)

where z > 0 denotes the dual variable to x.

The value of *x* which the maximum in the above equation will be attained at is denoted by g(t, s, z), so we have

$$g(t, s, z) = \inf_{x>0} \left\{ x \mid H(t, s, x) \ge zx + \widehat{H}(t, s, z) \right\}.$$
 (20)

The functions g(t, s, z) and  $\widehat{H}(t, s, z)$  are closely related and we will refer to either one of them as the dual function of H(t, s, x). In this paper, we will work mainly with the function g(t, s, z).

The function  $\widehat{H}(t, s, z)$  is related to g(t, s, z) by  $g(t, s, z) = -\widehat{H}_z(t, s, z)$ .

Noting that  $H(T, s, x) = (x - \gamma)^2$ , we can define the following Legendre transform at terminal time:

$$\widehat{H}(T, s, z) = \sup_{x>0} \left\{ \left( x - \gamma \right)^2 - zx \right\},$$

$$g(T, s, z) = \inf_{x>0} \left\{ \left( x \mid x - \gamma \right)^2 \ge zx + \widehat{H}(T, s, z) \right\}.$$
(21)

So we have

$$g(T, s, z) = \frac{1}{2}z + \gamma.$$
 (22)

According to (19), we have  $H_x(t, s, x) = z$ , and this leads to

$$g(t,s,z) = x, \qquad \widehat{H}(t,s,z) = H(t,s,g) - zg. \qquad (23)$$

Referring to Jonsson and Sircar [20], Xiao et al. [6], and Gao [7], we get the following transformation rules:

$$H_{t} = \widehat{H}_{t}, \qquad H_{x} = z, \qquad H_{xx} = -\frac{1}{\widehat{H}_{zz}},$$

$$H_{s} = \widehat{H}_{s}, \qquad H_{ss} = \widehat{H}_{ss} - \frac{\widehat{H}_{sz}^{2}}{\widehat{H}_{zz}}, \qquad H_{xs} = -\frac{\widehat{H}_{sz}}{\widehat{H}_{zz}}.$$
(24)

Putting transformation rules (24) into (18), we get

$$\begin{aligned} \widehat{H}_t + \mu s \widehat{H}_s + \frac{1}{2} k^2 s^{2\beta+2} \widehat{H}_{ss} + (\tau g) z \\ + \frac{(\mu - \tau)^2 z^2}{2k^2 s^{2\beta}} \widehat{H}_{zz} - (\mu - \tau) s z \widehat{H}_{sz} = 0. \end{aligned}$$
(25)

Differentiating  $\widehat{H}$  with respect to z, we derive the following dual equation:

$$g_{t} + \tau s g_{s} + \frac{1}{2} k^{2} s^{2\beta+2} g_{ss} + \left( \frac{(\mu - \tau)^{2}}{k^{2} s^{2\beta}} - \tau \right) z g_{z}$$

$$+ \frac{(\mu - \tau)^{2}}{2k^{2} s^{2\beta}} z^{2} g_{zz} - (\mu - \tau) s z g_{sz} - \tau g = 0,$$
(26)

where  $\tau = r$ , if  $(t, x) \in \Theta$ ;  $\tau = R$ , if  $(t, x) \notin \Theta$ .

Taking (22) into consideration, we can fit a solution to (26) with the following structure:

$$g(t, s, z) = \begin{cases} f^{r}(t, y) z + h^{r}(t), y = s^{-2\beta}, & \text{if } (t, x) \in \Theta; \\ f^{R}(t, y) z + h^{R}(t), y = s^{-2\beta}, & \text{if } (t, x) \notin \Theta. \end{cases}$$
(27)

Considering  $\tau = r$ , if  $(t, x) \in \Theta$ ;  $\tau = R$ , if  $(t, x) \notin \Theta$ , we can rewrite g(t, s, z) as

$$g(t,s,z) = f^{\tau}(t,y)z + h^{\tau}(t), \quad y = s^{-2\beta},$$
 (28)

with boundary conditions given by  $f^{\tau}(T, y) = 1/2$  and  $h^{\tau}(T) = \gamma$ .

Further, we have

$$g_{t} = f_{t}^{\tau} z + h_{t}^{\tau}, \qquad g_{s} = f_{y}^{\tau} (-2\beta) s^{-2\beta-1} z,$$

$$g_{sz} = f_{y}^{\tau} (-2\beta) s^{-2\beta-1},$$

$$g_{ss} = f_{yy}^{\tau} ((-2\beta) s^{-2\beta-1})^{2} \cdot z \qquad (29)$$

$$+ f_{y}^{\tau} (-2\beta) (-2\beta-1) s^{-2\beta-2} z,$$

$$g_{z} = f^{\tau}, \qquad g_{zz} = 0.$$

Putting the above partial derivatives into (26), we get

$$\begin{bmatrix} f_t^{\tau} + (2\beta(\mu - 2\tau)y + \beta(2\beta + 1)k^2)f_y^{\tau} \\ + 2\beta^2 k^2 y f_{yy}^{\tau} + \left(\frac{(\mu - \tau)^2}{k^2}y - 2\tau\right)f^{\tau}\end{bmatrix}z$$
(30)  
+  $h_t^{\tau} - \tau h^{\tau} = 0.$ 

Eliminating the dependence on *z*, we obtain

$$h_t^{\tau} - \tau h^{\tau} = 0, \qquad h(T) = \gamma; \tag{31}$$

$$f_{t}^{\tau} + \left(2\beta\left(\mu - 2\tau\right)y + \beta\left(2\beta + 1\right)k^{2}\right)f_{y}^{\tau} + 2\beta^{2}k^{2}yf_{yy}^{\tau} + \left(\frac{\left(\mu - \tau\right)^{2}}{k^{2}}y - 2\tau\right)f^{\tau} = 0, \qquad f^{\tau}\left(T, y\right) = \frac{1}{2}.$$
(32)

The solution to (31) is

$$h(t) = \gamma e^{-\tau(T-t)}.$$
(33)

**Lemma 3.** Assume that the structure of the solution to (32) is  $f^{\tau}(t, y) = A^{\tau}(t)e^{B^{\tau}(t)y}$ , with the boundary conditions given by  $A^{\tau}(T) = 1/2$  and  $B^{\tau}(T) = 0$ ; then  $A^{\tau}(t)$  and  $B^{\tau}(t)$  are given by (43) and (42), respectively.

Proof. Putting 
$$f^{\tau}(t, y) = A^{\tau}(t)e^{B^{\tau}(t)y}$$
 into (32), we have  

$$\begin{bmatrix} A^{\tau}(t) \frac{dB^{\tau}(t)}{dt} + 2\beta(\mu - 2\tau)A^{\tau}(t)B^{\tau}(t) \\ + 2\beta^{2}k^{2}A^{\tau}(t)B^{\tau^{2}}(t) + \frac{(\mu - \tau)^{2}}{k^{2}}A^{\tau}(t) \end{bmatrix} y \\ + \frac{dA^{\tau}(t)}{dt} + \beta(2\beta + 1)k^{2}A^{\tau}(t)B^{\tau}(t) - 2\tau A^{\tau}(t) = 0.$$
(34)

By matching the coefficients, we get

$$\frac{dA^{\tau}(t)}{dt} + \beta (2\beta + 1) k^{2} A^{\tau}(t) B^{\tau}(t)$$

$$- 2\tau A^{\tau}(t) = 0, \qquad A^{\tau}(T) = \frac{1}{2},$$

$$\frac{dB^{\tau}(t)}{dt} + 2\beta (\mu - 2\tau) B^{\tau}(t) + 2\beta^{2} k^{2} B^{\tau 2}(t)$$

$$+ \frac{(\mu - \tau)^{2}}{k^{2}} = 0, \qquad B^{\tau}(T) = 0.$$
(35)
(36)

Equation (36) can be reduced to

$$\frac{dB^{\tau}(t)}{dt} = -2\beta^2 k^2 B^{\tau^2}(t) - 2\beta \left(\mu - 2\tau\right) B^{\tau}(t) - \frac{\left(\mu - \tau\right)^2}{k^2}, \qquad B^{\tau}(T) = 0.$$
(37)

Let  $\Delta$  denote the discriminant of the quadratic equation  $-2\beta^2 k^2 B^{\tau 2}(t) - 2\beta(\mu - 2\tau)B^{\tau}(t) - ((\mu - \tau)^2/k^2) = 0.$ Easy calculation leads to  $\Delta = 4\beta^2(2\tau^2 - \mu^2)$ . Assume that

Easy calculation leads to  $\Delta = 4\beta^2(2\tau^2 - \mu^2)$ . Assume that  $\Delta > 0$ , that is,  $-\sqrt{2\tau} < \mu < \sqrt{2\tau}$ , then the quadratic equation has two real roots:

$$m_{1}^{\tau} = \frac{-(\mu - 2\tau) + \sqrt{2\tau^{2} - \mu^{2}}}{2\beta k^{2}},$$

$$m_{2}^{\tau} = \frac{-(\mu - 2\tau) - \sqrt{2\tau^{2} - \mu^{2}}}{2\beta k^{2}}.$$
(38)

So (37) can be rewritten as

$$\frac{1}{m_1^{\tau} - m_2^{\tau}} \int_t^T \left( \frac{1}{B^{\tau}(t) - m_1^{\tau}} - \frac{1}{B^{\tau}(t) - m_2^{\tau}} \right) dB^{\tau}(t)$$
  
=  $-2\beta^2 k^2 (T - t)$ . (39)

Further, we obtain

$$B^{\tau}(t) = \frac{m_1^{\tau} m_2^{\tau} \left(1 - e^{-2\beta^2 k^2 (m_1^{\tau} - m_2^{\tau})(T-t)}\right)}{m_1^{\tau} - m_2^{\tau} e^{-2\beta^2 k^2 (m_1^{\tau} - m_2^{\tau})(T-t)}}.$$
 (40)

Letting

$$\lambda_{1}^{\tau} = \frac{-(\mu - 2\tau) + \sqrt{2\tau^{2} - \mu^{2}}}{2\beta},$$
(41)

$$\lambda_{2}^{\tau} = \frac{-(\mu - 2\tau) - \sqrt{2\tau^{2} - \mu^{2}}}{2\beta}.$$

We get

$$B^{\tau}(t) = k^{-2}I^{\tau}(t),$$

$$I^{\tau}(t) = \frac{\lambda_{1}^{\tau}\lambda_{2}^{\tau}\left(1 - e^{-2\beta^{2}(\lambda_{1}^{\tau} - \lambda_{2}^{\tau})(T-t)}\right)}{\lambda_{1}^{\tau} - \lambda_{2}^{\tau}e^{-2\beta^{2}(\lambda_{1}^{\tau} - \lambda_{2}^{\tau})(T-t)}}.$$
(42)

Plugging (42) into (35) yields

$$A^{\tau}(t) = \frac{1}{2} e^{-\int_{t}^{T} (2\tau - \beta(2\beta + 1)I^{\tau}(t))dt}.$$
 (43)

Therefore, Lemma 3 is completed.

Finally, summarizing the above results, we obtain the optimal trading strategy for the problem (11).

**Theorem 4.** For a given  $\lambda$ , T and  $C \ge x_0 e^{rT}$ , the optimal investment strategy with borrowing constraint under a mean-variance criterion corresponding to the problem (11) is

$$\pi_{t}^{*} = \begin{cases} -\frac{(\mu - r)}{k^{2}S_{t}^{2\beta}} \left(X_{t} - \gamma e^{-r(T-t)}\right) \widetilde{K}^{r}(t), \\ if X_{t} \ge \gamma \rho(t), \quad r < \mu < \sqrt{2}r, \\ -\frac{(\mu - R)}{k^{2}S_{t}^{2\beta}} \left(X_{t} - \gamma e^{-R(T-t)}\right) \widetilde{K}^{R}(t), \\ if X_{t} < \gamma \rho(t), \quad R < \mu < \sqrt{2}R, \end{cases}$$
(44)

where  $K^{\tau}(t)$  and  $\rho(t)$  are given by (48) and (50), respectively.

*Proof*. Under the transformation rules, the optimal strategy (16) is derived as follows:

$$\pi_{t}^{*} = \frac{-(\mu - \tau)H_{x} - k^{2}s^{2\beta+1}H_{xs}}{k^{2}s^{2\beta}H_{xx}}$$

$$= \frac{-(\mu - \tau)H_{x}/H_{xx} - k^{2}s^{2\beta+1}H_{xs}/H_{xx}}{k^{2}s^{2\beta}}$$

$$= \frac{(\mu - \tau)z\widehat{H}_{zz} - k^{2}s^{2\beta+1}\widehat{H}_{sz}}{k^{2}s^{2\beta}}$$

$$= \frac{-(\mu - \tau)zg_{z} + k^{2}s^{2\beta+1}g_{s}}{k^{2}s^{2\beta}}.$$
(45)

Taking (33),(35), and Lemma 3 into consideration, we have

$$\begin{aligned} \pi_t^* &= \frac{-(\mu - \tau) zg_z + k^2 s^{2\beta + 1} g_s}{k^2 s^{2\beta}} \\ &= \frac{-(\mu - \tau) zf^\tau + k^2 s^{2\beta + 1} zf_y^\tau (-2\beta) s^{-2\beta - 1}}{k^2 s^{2\beta}} \\ &= \frac{-(\mu - \tau) (g - h^\tau (t)) + k^2 zA^\tau (t) B^\tau (t) e^{B^\tau (t) y} (-2\beta)}{k^2 s^{2\beta}} \\ &= \frac{-(\mu - \tau) (x - h^\tau (t)) + k^2 B^\tau (t) (-2\beta) (x - h^\tau (t))}{k^2 s^{2\beta}} \\ &= \frac{-(\mu - \tau) (x - h^\tau (t)) + I^\tau (t) (-2\beta) (x - h^\tau (t))}{k^2 s^{2\beta}} \\ &= -\frac{(\mu - \tau) + 2\beta I^\tau (t)}{k^2 s^{2\beta}} (x - h^\tau (t)). \end{aligned}$$

Therefore, the optimal strategy is reduced to

$$\pi_t^* = \begin{cases} -\frac{(\mu - r)}{k^2 S_t^{2\beta}} \left( X_t - \gamma e^{-r(T-t)} \right) \widetilde{K}^r(t), \\ \text{if } (t, x) \in \Theta, \ r < \mu < \sqrt{2}r, \\ -\frac{(\mu - R)}{k^2 S_t^{2\beta}} \left( X_t - \gamma e^{-R(T-t)} \right) \widetilde{K}^R(t), \\ \text{if } (t, x) \notin \Theta, \ R < \mu < \sqrt{2}R, \end{cases}$$
(47)

where

$$\widetilde{K}^{\tau}(t) = \left[1 + \frac{2\beta I^{\tau}(t)}{\mu - \tau}\right], \quad \tau = r, \text{ if } (t, x) \in \Theta,$$

$$\tau = R, \text{ if } (t, x) \notin \Theta.$$
(48)

As the boundary condition of borrowing the money from the bank is  $X_t - \pi_t^* = 0$ ; that is,

$$X_t + \frac{\left(\mu - R\right)}{k^2 S_t^{2\beta}} \left(X_t - \gamma e^{-R(T-t)}\right) \widetilde{K}^R(t) = 0.$$
(49)

Denoting by  $\gamma \rho(t)$  the borrowing curve, we yield

$$\rho(t) = \frac{\left(\left(\mu - R\right)/k^2 S_t^{2\beta}\right) e^{-R(T-t)} \widetilde{K}^R(t)}{1 + \left(\left(\mu - R\right)/k^2 S_t^{2\beta}\right) \widetilde{K}^R(t)}.$$
(50)

Therefore, nonborrowing region  $\Theta$  in the (t, x)-plane can be rewritten as

$$\Theta = \left\{ (t, x) \in [0, T] \times \mathbb{R} \mid X_t \ge \gamma \rho(t) \right\}.$$
(51)

Hence, the proof of the Theorem 4 is completed.  $\Box$ 

Remark 5. We can draw some conclusions from (44).

- (i) If X<sub>t</sub> ≥ γρ(t) and r < μ < √2r, the investor need not borrowing the money from the bank and the optimal amount invested in the stock can be calculated by the first equation of (44), while the amount invested in the bond is X<sub>t</sub> − π<sup>\*</sup><sub>t</sub>.
- (ii) If  $X_t < \gamma \rho(t)$  and  $R < \mu < \sqrt{2}R$ , the optimal amount invested in the stock is given by the second equation of (44). In addition, investors need to borrow the money to invest the stock and the amount to borrow is  $|X_t \pi_t^*|$ , while the bond need not be invested.

### 4. The Efficient Frontier

In this section, we apply Lagrange duality theorem to derive the efficient frontier for the mean-variance portfolio selection problem (7). To simplify the presentation, we denote by  $\tau$ either the interest rate *r* or the borrowing rate *R*, and letting

$$\theta^{\tau} = \frac{\mu - \tau}{k S_t^{\beta}},\tag{52}$$

where  $\tau = r$ , if  $X_t \ge \gamma \rho(t)$  and  $\tau = R$ , if  $X_t < \gamma \rho(t)$ .

In both cases above, the wealth equation (5) is reduced to

$$dX_{t} = (\tau X_{t} + (\mu - \tau) \pi_{t}) dt + \pi_{t} k S_{t}^{\beta} dW_{t}, \quad X_{0} = x_{0} > 0.$$
(53)

For any fixed  $\gamma$ , under the efficient strategy in the Theorem 4, the dynamics of the wealth equation (5) are

$$dX_{t} = \left(\left(\tau - \left(\theta^{\tau}\right)^{2}\widetilde{K}^{\tau}\left(t\right)\right)X_{t} + \left(\theta^{\tau}\right)^{2}\gamma e^{-\tau(T-t)}\widetilde{K}^{\tau}\left(t\right)\right)dt$$
$$-\theta^{\tau}\left(X_{t} - \gamma e^{-\tau(T-t)}\right)\widetilde{K}^{\tau}\left(t\right)dW_{t}, \quad X_{0} = x_{0} > 0.$$
(54)

Applying Itô's lemma to the wealth process (54), we yield

$$dX_{t}^{2} = \left( \left[ 2\tau + \left(\theta^{\tau}\right)^{2} \left( \widetilde{K}^{2\tau}\left(t\right) - 2\widetilde{K}^{\tau}\left(t\right) \right) \right] X_{t}^{2} - \left(\theta^{\tau}\right)^{2} 2\gamma e^{-\tau(T-t)} \left( \widetilde{K}^{2\tau}\left(t\right) - \widetilde{K}^{\tau}\left(t\right) \right) X_{t} + \left(\theta^{\tau}\right)^{2} \left( \gamma e^{-\tau(T-t)} \widetilde{K}^{\tau}\left(t\right) \right)^{2} \right) dt - \theta^{\tau} \left( X_{t} - \gamma e^{-\tau(T-t)} \right) \widetilde{K}^{\tau}\left(t\right) 2X_{t} dW_{t}, \quad X_{0}^{2} = x_{0}^{2} > 0.$$
(55)

Taking expectations on both sides of (54) and (55), respectively, one has

$$d\mathbb{E}X_{t} = \left(\left(\tau - \left(\theta^{\tau}\right)^{2}\widetilde{K}^{\tau}\left(t\right)\right)\mathbb{E}X_{t} + \left(\theta^{\tau}\right)^{2}\gamma e^{-\tau(T-t)}\widetilde{K}^{\tau}\left(t\right)\right)dt,$$
$$\mathbb{E}X_{0} = x_{0} > 0,$$

$$d\mathbb{E}X_{t}^{2} = \left(\left[2\tau + \left(\theta^{\tau}\right)^{2}\left(\widetilde{K}^{2\tau}\left(t\right) - 2\widetilde{K}^{\tau}\left(t\right)\right)\right]\mathbb{E}X_{t}^{2} - \left(\theta^{\tau}\right)^{2}2\gamma e^{-\tau\left(T-t\right)}\left(\widetilde{K}^{2\tau}\left(t\right) - \widetilde{K}^{\tau}\left(t\right)\right)\mathbb{E}X_{t} + \left(\theta^{\tau}\right)^{2}\left(\gamma e^{-\tau\left(T-t\right)}\widetilde{K}^{\tau}\left(t\right)\right)^{2}\right)dt, \quad \mathbb{E}X_{0}^{2} = x_{0}^{2} > 0.$$
(57)

The solution of the linear ordinary differential equation (56) is

$$\mathbb{E}X_t = x_0 e^{\int_0^t (\tau - (\theta^{\mathsf{r}})^2 \widetilde{K}^{\mathsf{r}}(t)) dt} + \gamma e^{-\tau (T-t)} \left[ 1 - e^{-\int_0^t (\theta^{\mathsf{r}})^2 \widetilde{K}^{\mathsf{r}}(t) dt} \right],$$
(58)

and it results in

$$\mathbb{E}X_T = x_0 e^{\int_0^T (\tau - (\theta^{\tau})^2 \widetilde{K}^{\tau}(t)) dt} + \gamma \left[ 1 - e^{-\int_0^T (\theta^{\tau})^2 \widetilde{K}^{\tau}(t) dt} \right].$$
(59)

Similarly, by solving (57), one has

$$\mathbb{E}X_T^2 = e^{\int_0^T (\theta^{\tau})^2 (\widetilde{K}^{2\tau}(t) - 2\widetilde{K}^{\tau}(t))dt} (x_0 e^{\tau T} - \gamma)^2 + 2\gamma e^{-\int_0^T (\theta^{\tau})^2 \widetilde{K}^{\tau}(t)dt} (x_0 e^{\tau T} - \gamma) + \gamma^2.$$
(60)

Therefore, the objective function of the problem (9), as a explicit function of parameter  $\gamma$ , is given by

$$\begin{split} \overline{J}_{\min}(\pi_{t}^{*},\gamma) &= \mathbb{E}(X_{T}-\gamma)^{2} - (C-\gamma)^{2} \\ &= \mathbb{E}X_{T}^{2} - 2\gamma \mathbb{E}X_{T} + 2\gamma C - C^{2} \\ &= \gamma^{2} \left( e^{\int_{0}^{T} (\theta^{\tau})^{2} (\widetilde{K}^{2\tau}(t) - 2\widetilde{K}^{\tau}(t)) dt} - 1 \right) \\ &+ 2\gamma \left( C - x_{0} e^{\tau T} e^{\int_{0}^{T} (\theta^{\tau})^{2} (\widetilde{K}^{2\tau}(t) - 2\widetilde{K}^{\tau}(t)) dt} \right) \\ &+ x_{0}^{2} e^{2\tau T} e^{\int_{0}^{T} (\theta^{\tau})^{2} (\widetilde{K}^{2\tau}(t) - 2\widetilde{K}^{\tau}(t)) dt} - C^{2}. \end{split}$$
(61)

Using (10) obtained by Lagrange duality theorem, the minimum variance  $\operatorname{Var} X_T$  is achieved for

$$\gamma_{\tau}^{*} = \frac{C - x_{0} e^{\tau T} e^{\int_{0}^{T} (\theta^{\tau})^{2} (\widetilde{K}^{2\tau}(t) - 2\widetilde{K}^{\tau}(t)) dt}}{1 - e^{\int_{0}^{T} (\theta^{\tau})^{2} (\widetilde{K}^{2\tau}(t) - 2\widetilde{K}^{\tau}(t)) dt}}.$$
 (62)

In addition, we obtain

$$\overline{J}_{\max\min}\left(\pi_{t}^{*},\gamma_{\tau}^{*}\right) = \frac{\left(C - x_{0}e^{\tau T}\right)^{2}}{e^{\int_{0}^{T}(\theta^{\tau})^{2}(\widetilde{K}^{2\tau}(t) - 2\widetilde{K}^{\tau}(t))dt} - 1}.$$
(63)

Letting  $\operatorname{Var}^{\tau} X_T = \overline{J}_{\max\min}(\pi_t^*, \gamma_{\tau}^*)$ . The optimal value of  $\gamma$  and the minimum variance  $\operatorname{Var} X_T$  are

$$\operatorname{Var} X_{T} = \operatorname{Max} \left\{ \operatorname{Var}^{r} X_{T}, \operatorname{Var}^{R} X_{T} \right\}$$
$$= \operatorname{Max} \left\{ \frac{\left( C - x_{0} e^{rT} \right)^{2}}{e^{\int_{0}^{T} (\theta^{r})^{2} (\widetilde{K}^{2r}(t) - 2\widetilde{K}^{r}(t)) dt} - 1},$$
(64)

$$\frac{\left(C - x_0 e^{RT}\right)^2}{e^{\int_0^T (\theta^R)^2 (\widetilde{K}^{2R}(t) - 2\widetilde{K}^R(t))dt} - 1} \right\},$$

$$\gamma^* = \begin{cases} \gamma_r, & \text{if } \text{Var } X_T = \text{Var } X_T, \\ \gamma_R^*, & \text{if } \text{Var}^R X_T = \text{Var } X_T. \end{cases}$$
(65)

Putting (65) into the optimal strategy (44) in Theorem 4, we can summarize our main results in the following theorem.

**Theorem 6.** The optimal investment strategy for the meanvariance portfolio selection problem (7) with borrowing constraint under a CEV process is

$$\pi_{t}^{*} = \begin{cases} -\frac{(\mu - r)}{k^{2}S_{t}^{2\beta}} \left(X_{t} - \gamma^{*}e^{-r(T-t)}\right) \widetilde{K}^{r}(t), \\ if X_{t} \ge \gamma^{*}\rho(t), \quad r < \mu < \sqrt{2}r, \\ -\frac{(\mu - R)}{k^{2}S_{t}^{2\beta}} \left(X_{t} - \gamma^{*}e^{-R(T-t)}\right) \widetilde{K}^{R}(t), \\ if X_{t} < \gamma^{*}\rho(t), \quad R < \mu < \sqrt{2}R. \end{cases}$$
(66)

Moreover, the efficient frontier is given by

$$\operatorname{Var} X_{T} = \operatorname{Max} \left\{ \frac{\left(C - x_{0}e^{rT}\right)^{2}}{e^{\int_{0}^{T} (\theta^{r})^{2}(\widetilde{K}^{2r}(t) - 2\widetilde{K}^{r}(t))dt} - 1}, \frac{\left(C - x_{0}e^{RT}\right)^{2}}{e^{\int_{0}^{T} (\theta^{R})^{2}(\widetilde{K}^{2R}(t) - 2\widetilde{K}^{R}(t))dt} - 1} \right\},$$
(67)

where  $K^{\tau}(t)$  and  $\rho(t)$  are given by (48) and (50), respectively.

*Remark 7.* When  $\beta = 0$ , the results in the Theorem 6 are reduced to the ones under a geometric Brownian motion model, which is obtained by [18]. When  $\beta = -1/2$  and  $\beta = -1$ , the corresponding results are all given by (66) and (67). Therefore, extending a geometric Brownian motion to a CEV model is the most important innovation in our paper.

### **5. Conclusions**

This paper is concerned with a continuous-time dynamic portfolio selection problem in a mean-variance framework, in which the constraint of the borrowing rate higher than the lending rate is allowed and stock price process is supposed to follow the constant elasticity of variance (CEV) model. The closed-form solution to the optimal investment strategy is derived by applying Legendre transform and dual theory. In addition, the efficient strategy and efficient frontier are derived by using Lagrange duality theorem.

In future research, we will continue to concentrate on continuous-time portfolio selection problems under a CEV model. It would be interesting to extend our model to those with more sophisticated cases, such as introducing consumption and transaction cost, short-selling constraint, and liability process. We leave these problems and corresponding verification theorem for future research.

#### Acknowledgments

This research is supported by the Humanities and Social Science Research Youth Foundation of Ministry of Education (no. 11YJC790006) and the Higher School Science and Technology Development Foundation of Tianjin (no. 20100821).

#### References

- J. C. Cox and S. A. Ross, "The valuation of options for alternative stochastic processes," *Journal of Financial Economics*, vol. 3, no. 1-2, pp. 145–166, 1976.
- [2] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637– 654, 1973.
- [3] M. Schroder, "Computing the constant elasticity of variance option pricing formula," *Journal of Finance*, vol. 44, no. 1, pp. 211–219, 1989.
- [4] P. B. Phelim and S. T. Yisong, "Pricing lookback and barrier options under the CEV process," *Journal of Financial and Quantitative Analysis*, vol. 34, no. 2, pp. 241–264, 1999.

- [5] D. Davydov and V. Linetsky, "Pricing and hedging pathdependent options under the CEV process," *Management Science*, vol. 47, no. 7, pp. 949–965, 2001.
- [6] J. Xiao, Z. Hong, and C. Qin, "The constant elasticity of variance (CEV) model and the Legendre transform-dual solution for annuity contracts," *Insurance:Mathematics and Economics*, vol. 40, no. 2, pp. 302–310, 2007.
- [7] J. Gao, "Optimal investment strategy for annuity contracts under the constant elasticity of variance (CEV) model," *Insurance:Mathematics and Economics*, vol. 45, no. 1, pp. 9–18, 2009.
- [8] J. Gao, "An extended CEV model and the Legendre transform-dual-asymptotic solutions for annuity contracts," *Insurance:Mathematics and Economics*, vol. 46, no. 3, pp. 511–530, 2010.
- [9] C. Paxson, "Borrowing constraints and portfolio choice," *The Quarterly Journal of Economics*, vol. 105, no. 2, pp. 535–543, 1990.
- [10] W. H. Fleming and T. Zariphopoulou, "An optimal investment/consumption model with borrowing," *Mathematics of Operations Research*, vol. 16, no. 4, pp. 802–822, 1991.
- [11] J.-L. Vila and T. Zariphopoulou, "Optimal consumption and portfolio choice with borrowing constraints," *Journal of Economic Theory*, vol. 77, no. 2, pp. 402–431, 1997.
- [12] L. Teplá, "Optimal portfolio policies with borrowing and shortsale constraints," *Journal of Economic Dynamics & Control*, vol. 24, no. 11-12, pp. 1623–1639, 2000.
- [13] T. Zariphopoulou, "Consumption-investment models with constraints," *SIAM Journal on Control and Optimization*, vol. 32, no. 1, pp. 59–84, 1994.
- [14] X. Y. Zhou and D. Li, "Continuous-time mean-variance portfolio selection: a stochastic LQ framework," *Applied Mathematics* and Optimization, vol. 42, no. 1, pp. 19–33, 2000.
- [15] X. Li, X. Y. Zhou, and A. E. B. Lim, "Dynamic mean-variance portfolio selection with no-shorting constraints," *SIAM Journal* on Control and Optimization, vol. 40, no. 5, pp. 1540–1555, 2002.
- [16] M. C. Chiu and D. Li, "Asset and liability management under a continuous-time mean-variance optimization framework," *Insurance:Mathematics and Economics*, vol. 39, no. 3, pp. 330– 355, 2006.
- [17] S. Xie, Z. Li, and S. Wang, "Continuous-time portfolio selection with liability: mean-variance model and stochastic LQ approach," *Insurance:Mathematics and Economics*, vol. 42, no. 3, pp. 943–953, 2008.
- [18] C. Fu, A. Lari-Lavassani, and X. Li, "Dynamic mean-variance portfolio selection with borrowing constraint," *European Journal of Operational Research*, vol. 200, no. 1, pp. 312–319, 2010.
- [19] D. G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, NY, USA, 1968.
- [20] M. Jonsson and R. Sircar, "Optimal investment problems and volatility homogenization approximations," in *Modern Methods* in Scientific Computing and Applications, vol. 75 of NATO Science Series II, pp. 255–281, Springer, Berlin, Germany, 2002.

### Research Article

# A Fast Optimization Method for Reliability and Performance of Cloud Services Composition Application

### Zhao Wu,<sup>1</sup> Naixue Xiong,<sup>2</sup> Yannong Huang,<sup>1</sup> Qiong Gu,<sup>1</sup> Chunyang Hu,<sup>1</sup> Zhongbo Wu,<sup>1</sup> and Bo Hang<sup>1</sup>

<sup>1</sup> School of Mathematics and Computer Science, Hubei University of Arts and Science, Xiangyang 441053, China <sup>2</sup> School of Computer Science, Colorado Technical University, Colorado Springs, CO 80907, USA

Correspondence should be addressed to Naixue Xiong; nxiong@coloradotech.edu

Received 15 April 2013; Accepted 13 September 2013

Academic Editor: Rung Ching Chen

Copyright © 2013 Zhao Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

At present the cloud computing is one of the newest trends of distributed computation, which is propelling another important revolution of software industry. The cloud services composition is one of the key techniques in software development. The optimization for reliability and performance of cloud services composition application, which is a typical stochastic optimization problem, is confronted with severe challenges due to its randomness and long transaction, as well as the characteristics of the cloud computing resources such as openness and dynamic. The traditional reliability and performance optimization techniques, for example, Markov model and state space analysis and so forth, have some defects such as being too time consuming and easy to cause state space explosion and unsatisfied the assumptions of component execution independence. To overcome these defects, we propose a fast optimization method for reliability and performance of cloud services composition application based on universal generating function and genetic algorithm in this paper. At first, a reliability and performance model for cloud service composition application based on the multiple state system theory is presented. Then the reliability and performance definition based on universal generating function is proposed. Based on this, a fast reliability and performance optimization algorithm is presented. In the end, the illustrative examples are given.

### 1. Introduction

Cloud computing is an emerging trend for the provision of IT infrastructure as services, with the potential of transforming the way of offering business services [1]. Based on cloud computing platform, software development becomes prominent and accessible for all without the expensive investing in hardware resources and the managing and maintaining costs.

On cloud computing platform, the cloud services composition (CSC) is a fashionable approach of software development based on cloud services [2–4]. In the framework of CSC, cloud services are considered as self-contained, self-describing, modular applications that can be published, located, and invoked across the web.

How to select and integrate cloud services to satisfy user's functional requirements is an important issue, which has widely attracted attention of researchers [5]. Great progress has been made in this field [6-8]. However, little research

focused on reliability model and simulation for CSC. Recently, there has been growing interest in this field. Methods and technologies related to reliability model and simulation for CSC have attracted attention because they can forecast the QoS that users will obtain from CSC [9–11]. In addition, it is helpful to analyze whether there are some reliability bottlenecks within CSC applications. Thus, reliability prediction is the basis of reliability optimization for the CSC applications.

The service-oriented architecture (SOA) is the most representative technological architecture to build the cloud services application on cloud computing platform [12–14]. However, because SOA supposed by services composition technique is of dynamic and cooperative essential characteristic, the traditional software reliability prediction methods are not suitable to the cloud services application based on SOA.

From the aspect of software architecture, cloud services application is a kind of Internetware based on cloud services,

which is built by cloud services composition technique [15, 16]. As a kind of abstract of distributed software system running on the Internet which is opened, dynamic, and difficult to control, there are many differences between the Internetware and traditional software system, such as structure, operation mechanism, correctness guarantees, development method, and life cycle. Due to the static, closed, and controllable running environment, the traditional software model is of finite autonomy, fixed encapsulation, monotonic interaction, tightly coupled structure, and offline evolution. Being different from the traditional software model, the cloud services application, as a kind of Internetware, exists in each node on the cloud service platform with a subjective software service form. In the running environment, which is opened, dynamic, and difficult to control, the cloud services application has some new characters, such as flexible evolution, continuous reaction, and multitarget self-adaption. Due to being difficult to adapt to these new characters, traditional software reliability assurance methods cannot be adopted for the cloud services application which is built based on service composition technique. Quite different to traditional software reliability assurance technique, the reliability assurance method for the cloud services application pays more attention to the mechanism of flexible reliability measure, predication and self-adapting based on summative evaluation of operation information in opened running environment [17, 18]. So, the fast reliability prediction method for the cloud services application has great theory research value.

From the aspect of software online evolution, cloud services application confronts fast and continuous change of user's requirement and running environment. So, cloud services application must have the ability to apperceive the changes in outrunning environment and dynamically evolve according to functionality and performance requirement with this kind of change. In order to provide better QoS to users, cloud services application must have more adaptability to collect various changes realtimely and adjust oneself online according to preestablished strategies in runtime [19]. However, with the closed, controllable, and static user's requirement in the background, traditional software reliability prediction methods lack the ability to dynamically adapt themselves to the changes of running environment and user's requirement. Therefore, it cannot be employed in the reliability prediction for cloud services application. So, the fast reliability prediction method for the cloud services application has important realistic technology requirement.

At the present time, the researches on reliability prediction for cloud services application are still just starting Due to the opened and dynamic running environment, continuous variable user's requirement, randomly selected member services and its own characters of loose coupling and long transaction, the severe challenges are confronted the reliability prediction for cloud services application, which is seriously restricting the further development, application, and extension of cloud services application. In the face of urgent demands of high reliable cloud services application from many government, economy, and commerce fields such as e-government, e-commerce, and e-bank, the fast reliability optimization becomes the key to promote the successful development, application, and extension of the cloud services application.

Facing the challenge, this paper researches the reliability model of cloud services application. On this basis, a fast reliability optimization for cloud services application is presented. The paper is organized as follows. Section 2 presents the reliability model for cloud service application based on the multiple state system (MSS) theory. The reliability and performance of cloud service and cloud services composition application are defined in Section 3. Section 4 presents a reliability and performance model for cloud services composition application based on universal generating function (referred to as UGF) technique. A fast reliability optimization algorithm by using the UGF technique is presented in Section 5. Section 6 provides some illustrative examples.

### 2. Reliability Model for Cloud Service Composition Application

2.1. Multiple State System Theory. The MSS was introduced in the middle of the 1970s in [20–23]. In these works, the basic concepts of MSS reliability were primarily formulated, the system structure function was defined, and its properties were initially studied. The notions of minimal cut set and minimal path set were introduced in the MSS context, as well as the notions of coherence and element relevancy.

Some systems can perform their tasks with various distinguished levels of efficiency usually referred to as performance rates. A system that can have a finite number of performance rates is called a multistate system. Any system consisting of different units that have a cumulative effect on the entire system performance has to be considered as a MSS [24]. So the cloud service application can be regard as a multiple state system.

MSS reliability analysis relates to systems for which one cannot formulate an "all or nothing" type of failure criterion. Such systems are able to perform their task with partial performance (intensity of the task accomplishment). Failures of some system elements lead only to the degradation of the system performance.

2.2. Reliability and Performance Definition for Cloud Services Composition Application. From the aspect of users, the reliability of cloud service application can be defined as the probability that its performance rates satisfy user's requirements, described as a vector pair  $(\mathbf{w}, \mathbf{q})$ , where  $\mathbf{w} = \{w_1, w_2, \ldots, w_M\}$  is a vector of user's requirement rates  $w_j$ ,  $(j = 1, \ldots, M)$ , and  $\mathbf{q} = \{q_1, q_2, \ldots, q_M\}$  is the vector of steady state probability  $q_j = \Pr\{W = w_j\}, (j = 1, \ldots, M)$ , according to a certain user's requirement rate, where W is a random variable that represents the performance rates of cloud service application.

Based on the above definition, the reliability function of cloud service application under steady state can be defined as

$$R(t) = \Pr\left\{T_f \ge t \mid F(G(0), W(0)) \ge 0\right\}.$$
 (1)

And the one under transient state can be defined as

$$R(t) = \Pr\{F(G(t), W(t)) \ge 0\},$$
(2)

where G(t) is the integral performance rate of cloud service application.

In the interval [0, T], the reliability function of cloud service application can be defined as

$$R_T = \frac{1}{T} \int_0^T 1\left(F\left(G\left(t\right), W\left(t\right)\right) \ge 0\right) dt.$$
 (3)

Based on the above definition, the reliability function of cloud service application under dynamically changing user's requirements can be defined as

$$R(\mathbf{w}, \mathbf{q}) = \sum_{m=1}^{M} R(w_m) q_m$$

$$= \sum_{m=1}^{M} q_m \sum_{k=1}^{K} p_k 1(F(g_k, w_m) \ge 0).$$
(4)

In order to calculate the probability distribution of reliability, failure time  $T_f$ , time between failures  $T_b$ , and failure number  $N_T$  are defined.

2.3. Probability Distribution of Performance Rates for Cloud Service. Furthermore, the performance rates for cloud service can be defined. According to its performance rates, the cloud service *j* to build a cloud service application can be of  $k_j$ kinds of various states, described by  $\mathbf{g}_j = \{g_{j1}, g_{j2}, \dots, g_{jk_j}\}$ , where  $g_{ji}$  is the performance rate of cloud service *j* under the state *i*,  $i \in \{1, 2, \dots, k_j\}$ . The performance rate  $G_j(t)$ corresponding to the cloud service *j* in any time  $t \ge 0$  is a random variable that gets the value from  $\mathbf{g}_j$ :  $G_j(t) \in \mathbf{g}_j$ . The probability of performance rates of the cloud service *j* under various states in any time *t* can be described as a set,

$$\mathbf{p}_{j}(t) = \left\{ p_{j1}(t), g_{j2}(t), \dots, g_{jk_{j}}(t) \right\},$$
(5)

where  $p_{ji}(t) = \Pr\{G_j(t) = g_{ji}\}\)$ . Because cloud service *j* is in one and only one of  $k_j$  kinds of various states in any time *t*, these states form a mutual exclusion events complete set. Therefore, the formula  $\sum_{i=1}^{k_j} p_{ji}(t) = 1$ ,  $(0 \le t \le T)$  is satisfied. In the end, the set of value pairs  $\langle g_{ji}, p_{ji}(t) \rangle$  completely determines the probability distribution of performance rates corresponding to a cloud service *j* in any time *t*.

2.4. Structure Function of Performance Rates for Cloud Service Application. Based on the above definition of the performance rates of cloud service application and cloud service, the structure function of cloud service application can be defined. Let

$$L^{n} = \{g_{11}, \dots, g_{1k_{1}}\} \times \{g_{21}, \dots, g_{2k_{2}}\} \times \dots \times \{g_{n1}, \dots, g_{nk_{n}}\}$$
(6)

be the possible combinations of performance rates of all cloud services and  $M = \{g_1, \ldots, g_k\}$  the possible values range of performance rates of cloud service application. Then the transform function  $\phi(G_1(t), \ldots, G_n(t)) : L^n \to M$ , called the structure function of cloud service application, can map the performance rates space of cloud services into one of cloud service applications. Hence, a general reliability model of cloud service application can be defined as

$$\mathbf{g}_{j}, \mathbf{p}_{j}(t), \quad 1 \le j \le n, \quad \phi\left(G_{1}(t), \dots, G_{n}(t)\right).$$
 (7)

The structure function of cloud service application establishes a feasible way to calculate the reliability of cloud service application using one of cloud services.

### 3. Reliability and Performance Definition for Cloud Service Composition Application Based on UGF

*3.1. UGF Technique.* The methods of MSS reliability assessment are based on four different approaches: (1) an extension of the Boolean models to the multivalued case; (2) the stochastic process (mainly Markov and semi-Markov) approach; (3) the Monte-Carlo simulation technique; and (4) the UGF approach.

The approach based on the extension of Boolean models is historically the first method that was developed and applied for the MSS reliability evaluation. It is based on the natural expansion of the Boolean methods to the multistate systems.

The stochastic process methods that are widely used for the MSS reliability analysis are more universal. The method can be applied only to relatively small MSS because the number of system states increases dramatically with the increase in the number of system elements.

Even though almost every real world MSS can be represented by the Monte-Carlo simulation for the reliability assessment, the main disadvantages of this approach are the time and expenses involved in the development and execution of the model.

The computational burden is the crucial factor when one solves optimization problems where the reliability measures have to be evaluated for a great number of possible solutions along the search process. This makes using the three abovementioned methods in reliability optimization problematic. On the contrary, the UGF technique is fast enough. This technique allows one to find the entire MSS performance distribution based on the performance distribution of its elements by suing a fast algebraic procedure. An analyst can use the same recursive procedures for MSS with a different physical nature of performance and different types of element interaction.

For the above reasons, we choose UGF technique to study a fast reliability optimization method for cloud services composition network. The UGF generalizes the technique that is based on using a well-known ordinary generating function. The basic ideas of the method were introduced by Ushakov [25]. The approach proved to be very convenient for numerical realization. It requires relatively small computational resources for evaluating MSS reliability indices and, therefore, can be used in complexes reliability optimization algorithms. Because the relationship between system state probability and system output performance rates can be expressed definitely by UGF, and the *u*-function of system can be obtained by calculating the *u*-function of components simply, UGF is approved as an efficient reliability assessment approach that is suitable to various MMS. Therefore, UGF can be successfully applied for the reliability assessment and optimization of MMS.

In most studies on the prediction and optimization of system reliability based on the UGF, the system structure and composite form of research object are relatively simple, such as electric power system and mechanical system. The presented design methods and calculation methods of the *u*-function composite operators are only applicable to some simple structure forms, such as series, parallel, series parallel hybrid, and bridge structure, which limits the application range of the UGF method. Different from the above research objects, the cloud services composition is of complex, flexible and dynamic structure form. The adaptability to complex, flexible and dynamic system, such as the cloud services composition application, becomes advantage and characteristic of our presented method.

3.2. Reliability Definition of Cloud Service Composition Application Based on UGF. Based on the reliability model for cloud service application described in Section 2, the reliability of cloud service composition can be defined by UGF. The general form of the definition is as follows.

The reliability of a cloud service composition (or a cloud service) is a random variable *X*. Therefore, the corresponding *u*-function can be defined as

$$u(z) = \sum_{k=1}^{K} p_k \cdot z^{X_k},$$
(8)

where the discrete variable *X* has *K* possible values and  $p_k$  is the reliability when *X* is in the state  $X_k$ . Based on this definition, the reliability of a cloud service composition (or a cloud service) can be expressed as

$$U(t,z) = \sum_{k=1}^{K} p_k(t) \cdot z^{G_k}.$$
 (9)

Because U(z) is correlative with the state probability  $p_k$ and the reliability rate  $G_k$ , which correspond to the cloud service composition (or a cloud service), describes the reliability of cloud service composition (or a cloud service). On this basis, we can define related performance operators furthermore, such as usability operator  $\delta_A$ , output performance operator  $\delta_G$  and unfinished performance operator  $\delta_U$ , to describe related reliability indexes.

Based on the above performance operators, the related reliability indexes for cloud service composition (or a cloud service) can be defined as follows.

(i) The usability is defined as

$$E_A = E_A(W,q) = \sum_{m=1}^{M} q_m \cdot \delta_R(U(z), F, W_m).$$
(10)

(ii) The output performance expectation is defined as

$$E_{G} = \delta_{G}(U(z)) = \delta_{G}\left(\sum_{k=1}^{K} p_{k} \cdot z^{G_{k}}\right)$$

$$= \frac{dU}{dz}(1) = \sum_{k=1}^{K} p_{k} \cdot G_{k}.$$
(11)

(iii) The unfinished performance requirement is defined as:

$$E_{U}(W,q) = \sum_{m=1}^{M} q_{m} \cdot \delta_{U}(U(z), F, W_{m}), \qquad (12)$$

where

$$\delta_{U}\left(U\left(z\right), F, W_{m}\right)$$

$$= \delta_{U}\left(\sum_{k=1}^{K} p_{k} \cdot z^{G_{k}}, F, W_{m}\right)$$

$$= \sum_{k=1}^{K} p_{k} \cdot \max\left\{-F\left(G_{k}, W_{m}\right), 0\right\}.$$
(13)

3.3. Composite Operators of Reliability and Performance Indexes Based on UGF. Based on the above reliability definition expressed by UGF for cloud services, the *u*-function composite operators  $\Omega$  can be designed for various performance indexes of the diverse composition patterns. The reliability of cloud service composition can be worked out based on the  $\Omega$  calculation of cloud services' reliability.

Two rules must be satisfied in the design of *u*-function composite operators  $\Omega$  as follows:

(1) 
$$\Omega (U_1 (z), ..., U_k (z), U_{k+1} (z), ..., U_n (z))$$
  
 $= \Omega (U_1 (z), ..., U_{k+1} (z), U_k (z), ..., U_n (z));$   
(2)  $\Omega (U_1 (z), ..., U_k (z), U_{k+1} (z), ..., U_n (z))$   
 $= \Omega (\Omega (U_1 (z), ..., U_k (z)), \Omega (U_{k+1} (z), ..., U_n (z))),$   
(14)

The generic form of composite operators  $\boldsymbol{\Omega}$  can be expressed as

$$\Omega\left(\sum_{\forall k} p_k \cdot z^{G_k}, \sum_{\forall l} p_l \cdot z^{G_l}\right) = \sum_{\forall k} \sum_{\forall l} p_k \cdot p_l \cdot z^{f(G_k, G_l)}, \quad (15)$$

where  $f(G_k, G_l)$  can be defined according to the performance indexes and composition structures of the cloud service application.

### 4. Reliability and Performance Model of Cloud Services Composition Application Based on UGF

4.1. Fault Tolerant Model of Cloud Services Composition Application. To strengthen the capability of fault tolerant of cloud service composition applications to improve its reliability and performance, the component duplication technique has been introduced into the design of cloud service composition applications. By deploying a number of functionally equivalent software versions for each cloud service, the cloud service composition applications can avoid the global failure due to the fault of one cloud service as far as possible. We assume that  $n_c$  functionally equivalent software versions are available for each cloud service *c*. Each software version *i* has an estimated reliability  $r_{ci}$  and response time  $\tau_{ci}$  (it includes the execution time of software version and the network transmission time used transfer computing results to other software version, end users, etc.). Failures of software versions in each cloud service are statistically independent, as well as the total failures of the different cloud services.

In many cases, the information about the software version's reliability and the response time is available from separate testing and/or reliability prediction models. This information can be incorporated into a fault-tolerant program model in order to obtain an evaluation of its reliability and performance.

According to the generally accepted model, the cloud service composition application consists of C cloud services. Each cloud service performs a subtask and the sequential execution of the cloud services performs a major task.

To assure that all of the computing tasks can correctly be executed by cloud service, the cloud services broker (referred to as CSB) and the check mechanism are established in the cloud service composition application. The check mechanism presumes that software versions send their computing results to the CSB. Then the CSB compares received computing results with each other. The CSB sends the computing results to the next cloud service in the service flow, if at least *k* out of *n* computing results agree. Otherwise, the CSB discards these received computing results and recalls the cloud services. If the consistent results cannot be obtained after trying a certain number of times, the CSB will stop the execution of the cloud services composition application.

The software versions in each cloud service c run on parallel hardware units. The total number of units is  $h_c$ . The units are independent and identical. The availability of each unit is  $a_c$ . The number  $H_c$  of units available at the moment determines the amount of available computational resources and, therefore, the number of software versions that can be executed simultaneously  $L_c(H_c)$ . No hardware unit can change its state during the execution.

The software versions of each cloud service *c* start their execution in accordance with a predetermined ordered list.  $L_c$  first software versions from the list start their execution simultaneously (at time zero). If the number of terminated software versions is less than  $k_c$ , after termination of each software version a new software version from the list starts its execution immediately. If the number of terminated software versions is not less than  $k_c$ , after termination of each software version the CSB compares the outputs. If  $k_c$  outputs are identical, the CSB terminates its execution (terminating all the software versions that are still executed), otherwise a new software version from the list is executed immediately.

If after termination of  $n_c$  software versions the number of identical outputs is less than  $k_c$  then the entire cloud services application fail.

The execution time of the CSB includes the execution time and data transmission time spent by all cloud services invoked by the CSB and itself. In the case that CSB gets not less than  $k_c$  consistent results successfully, the time of the entire CSB execution  $T_c$  is equal to the termination time of the software version that has produced the  $k_c$ th correct output (in most cases, the time needed by the CSB to make the decision can be neglected). It can be seen that the CSB execution time is a random variable depending on the reliability and the response time of the software versions and on the availability of the hardware units.

The sum of the random execution times of each CSB gives the random task execution time for the entire system T. In order to estimate both the system's reliability and its performance, different measures can be used, depending on the application.

In cloud service applications where the response time of each task is of critical importance, the system's acceptability function is defined as F(T, w) = 1(T < w), where w is a maximal allowed system response time. The system's reliability R(w) = E(F(T, w)) in this case is the probability that the correct output is produced in time less than w. The conditional expected system response time  $\tilde{\epsilon}(w) = E(T \times 1(T < w))/R(w)$ is considered to be a measure of the system's performance. This index, defined according to (16), determines the system's expected response time given that the system and network do not fail:

$$\widetilde{\varepsilon}(w) = \frac{E(\widetilde{G})}{\Pr\{F(G,W)=1\}} = \frac{E(GF(G,W))}{E(F(G,W))}.$$
 (16)

In cloud service applications where the system's average productivity (the number of executed tasks) over a fixed mission time is of interest, the system's acceptability function is defined as  $F(T) = 1(T < \infty)$ , the system's reliability is defined as the probability that it produces correct outputs regardless of the total response time (this index can be referred to as  $R(\infty)$ ), and the conditional expected system response time  $\tilde{\epsilon}(\infty)$  is considered to be a measure of the system's performance.

4.1.1. Number of Software Versions That Can Be Simultaneously *Executed*. The number of available hardware units in cloud service *c* can vary from 0 to  $h_c$ . Given that all of the units are identical and have availability  $a_c$ , one can easily obtain probabilities  $Q_c(x) = \Pr{H_c = x}$  for  $0 \le x \le h_c$ :

$$Q_{c}(x) = \Pr\{H_{c} = x\} = {\binom{h_{c}}{x}} a_{c}^{x} (1 - a_{c})^{h_{c} - x}.$$
 (17)

The number of available hardware units *x* determines the number of software versions that can be executed simultaneously:  $l_c(x)$ . Therefore,

$$\Pr\{L_{c} = l_{c}(x)\} = Q_{c}(x).$$
(18)

The pairs  $Q_c(x)$ ,  $l_c(x)$  for  $0 \le x \le h_c$  determine the probability mass function (referred to as p.m.f. as follows) of the discrete random value  $L_c$ .

4.1.2. Termination Times of Software Version. In each cloud service c, a sequence where each software version starts its execution is defined by the numbers of software versions. This means that each software version i starts its execution not earlier than software versions  $1, \ldots, i-1$  and not later than software versions  $i+1, \ldots, n_c$ . If the number of software versions that can run simultaneously is  $l_c$ , then we can assume that the software versions run on  $l_c$  independent processors. Let  $\alpha_m$  be the time when processor m terminates the execution of a software version and is ready to run the next software versions. Having the response time of each software version  $\tau_{ci}(1 \le i \le n_c)$ , one can obtain the termination time  $t_{ci}(l_c)$  for each software version i using the following simple algorithm.

- (1) Assign  $\alpha_1 = \cdots = \alpha_{l_c} = 0$  (all of the units are ready to run the software versions at time 0).
- (2) For  $i = 1, ..., n_c$  repeat the following:
  - (a) find any m (1 ≤ m ≤ l<sub>c</sub>) : α<sub>m</sub> = min{α<sub>1</sub>,..., α<sub>l<sub>c</sub></sub>} (m is the number of the earliest processor that is ready to run a new software version from the list),
  - (b) obtain  $t_{ci}(l_c) = \alpha_m + \tau_{ci}$  and assign  $\alpha_m = t_{ci}(l_c)$ .

Times  $t_{ci}(l_c)$ ,  $(1 \le i \le n_c)$ , correspond to intervals between the beginning of cloud service execution and the moment when the software versions produce their outputs. Observe that the software versions that start execution earlier can terminate later: j < y does not guarantee that  $t_{cj}(l_c) \le t_{cy}(l_c)$ . In order to obtain the sequence, in which the software versions produce their outputs, the termination times should be sorted in increasing order  $t_{cm_1}(l_c) \le t_{cm_2}(l_c) \le \cdots \le t_{cm_{n_c}}(l_c)$  which gives the order of software versions  $m_1, m_2, \ldots, m_{n_c}$  corresponding to times of their termination.

The ordered list  $m_1, m_2, \ldots, m_{n_c}$  determines the sequence of software version outputs. Now one can consider the cloud service c as a system in which the  $n_c$  software versions are executed consecutively according to the order  $m_1, m_2, \ldots, m_{n_c}$  and produce their outputs at times  $t_{cm_1}(l_c)$ ,  $t_{cm_2}(l_c), \ldots, t_{cm_{n_c}}(l_c)$ .

4.2. Definition of Reliability and Performance of the Cloud Service and the Cloud Services Composition Application. Let  $r_{cm_i}$  be the reliability of the software version that produces *i*th output in cloud service *c* ( $r_{cm_i}$  is equal to the probability that this output is correct). Consider the probability that *k* out of *n* first software versions of cloud service *c* succeed. This probability can be obtained as

$$R_{k} = \left[\prod_{i=1}^{n} \left(1 - r_{cm_{i}}\right)\right] \times \left[\sum_{i_{1}=1}^{n-k+1} \frac{r_{cm_{i_{1}}}}{1 - r_{cm_{i_{1}}}} \sum_{i_{2}=i_{1}+1}^{n-k+2} \frac{r_{cm_{i_{2}}}}{1 - r_{cm_{i_{2}}}} \cdots \sum_{i_{k}=i_{k-1}+1}^{n} \frac{r_{cm_{i_{k}}}}{1 - r_{cm_{i_{k}}}}\right].$$
(19)

The cloud service *c* produces the correct output directly after the end of the execution of *j* software versions  $(j \ge k_c)$  if the  $m_j$ th software version succeeds and exactly  $k_c - 1$  out of the first executed j - 1 software versions succeed.

The probability of such event  $p_{ci}(l_c)$  is

$$p_{cj}(l_{c}) = r_{cm_{j}} \left[ \prod_{i=1}^{j-1} \left( 1 - r_{cm_{i}} \right) \right] \\ \times \left[ \sum_{i_{1}=1}^{n-k_{c}+1} \frac{r_{cm_{i_{1}}}}{1 - r_{cm_{i_{1}}}} \sum_{i_{2}=i_{1}+1}^{n-k_{c}+2} \frac{r_{cm_{i_{2}}}}{1 - r_{cm_{i_{2}}}} \right] \cdots \sum_{i_{k_{c}}-1=i_{k_{c}-2}+1}^{j-1} \frac{r_{cm_{i_{k_{c}-1}}}}{1 - r_{cm_{i_{k_{c}-1}}}} \right].$$
(20)

Observe that  $p_{cj}(l_c)$  is the conditional probability that the cloud service response time is  $t_{cm_j}(l_c)$  given that  $l_c$  software versions can be executed simultaneously:

$$p_{cj}(l_c) = \Pr\left\{T_c = t_{cm_j}(l_c) \mid L_c = l_c\right\}.$$
 (21)

Having the p.m.f. of  $L_c$  we can now obtain for  $1 \le x \le h_c$ 

$$\Pr \left\{ T_{c} = t_{cm_{j}} \left( l_{c} \left( x \right) \right) \right\}$$
  
= 
$$\Pr \left\{ T_{c} = t_{ck_{j}} \left( l_{c} \left( x \right) \right) \mid L_{c} = l_{c} \left( x \right) \right\} \Pr \left\{ L_{c} = l_{c} \left( x \right) \right\}$$
  
= 
$$p_{cj} \left( l_{c} \left( x \right) \right) Q_{c} \left( x \right).$$
(22)

The pairs  $t_{cm_j}(l_c(x))$ ,  $p_{cj}(l_c(x))Q_c(x)$ , obtained for  $1 \le x \le h_c$  and  $k_c \le j \le n_c$ , determine the p.m.f. of software version response time  $T_c$ .

Since the events of successful cloud service execution termination for different j and x are mutually exclusive, we can express the probability of cloud service c success as

$$R_{c}(\infty) = \Pr\{T_{c} < \infty\} = \sum_{x=1}^{h_{c}} \left[Q_{c}(x)\sum_{j=k_{c}}^{n_{c}} p_{cj}(l_{c}(x))\right].$$
(23)

Since failure of any cloud service constitutes the failure of the entire application, the application's reliability can be expressed as

$$R(\infty) = \prod_{c=1}^{C} R_c(\infty).$$
(24)

For cloud services, there are four kinds of execution patterns in cloud services composition application: sequence, parallel, split, and loop. From the p.m.f. of response times  $T_c$  for each cloud service c one can obtain the p.m.f. of the response time of the entire application in accordance with

the composition structures and execution logics of the cloud services composition application:

$$T = \begin{cases} \sum_{c=1}^{C} T_c & \text{for sequence structure,} \\ \max(T_1, \dots, T_c) & \text{for parallel structure,} \\ \sum_{c=1}^{C} p_c T_c & \text{for split structure,} \\ \sum_{l=1}^{L} \sum_{c=1}^{C} T_c & \text{for loop structure,} \end{cases}$$
(25)

where the  $p_c$  is the probabilities that the *c*th split is chosen to be executed. The *L* is the number of times of loop that *C* cloud services are executed repeatedly.  $p_c$  and *L* can be obtained from separate testing and/or prediction models.

### 5. Fast Optimization Algorithm of Reliability and Performance for Cloud Services Composition Application Based on UGF and GA

5.1. Using UGF to Evaluate the Response Time Distribution of Cloud Services. In order to obtain the response time distribution for a cloud service c for a given  $l_c$  in the form  $p_{cj}(l_c)$ ,  $t_{cm_j}(l_c)$  ( $k_c \leq j \leq n_c$ ) one can determine the realizations  $t_{cm_j}(l_c)$  of the response time  $T_c(l_c)$  using the algorithm presented in Section 4.1.2 and the corresponding probabilities  $p_{cj}(l_c)$  using (20). However, the probabilities  $p_{cj}(l_c)$  can be obtained in a much simpler way using a procedure based on the UGF technique.

Let the random binary variable  $s_{cm_i}$  be an indicator of the success of software version  $m_i$  in cloud service c such that  $s_{cm_i} = 1$  if the software version produces the correct output and  $s_{cm_i} = 0$  if it produces the wrong output. The p.m.f. of  $s_{cm_i}$  can be represented by the *u*-function

$$u_{cm_i}(z) = r_{cm_i} z^1 + (1 - r_{cm_i}) z^0.$$
(26)

It can be easily seen that using the operator  $\otimes_+$  we can obtain the *u*-function

$$U_{cj}(z,l_c) = \bigotimes_{+} \left( u_{cm_i}(z), \dots, u_{cm_j}(z) \right)$$
(27)

that represents the p.m.f. of the number of correct outputs in cloud service *c* after the execution of a group of first *j* software versions (the order of elements  $m_1, m_2, \ldots, m_{n_c}$  and, therefore,  $U_{cj}(z, l_c)$  depend on  $l_c$ ). Indeed, the resulting polynomial relates the probabilities of combinations of correct and wrong outputs (the product of corresponding probabilities) with the number of correct outputs in these combinations (the sum of success indicators). Observe that after collecting the like terms (corresponding to obtaining the overall probability of

$$U_{cj}(z, l_c) = \sum_{k=0}^{j} \pi_{jk} z^k,$$
 (28)

where  $\pi_{jk}$  is the probability that the group of first *j* software versions produces *k* correct outputs.

Note that  $U_{cj}(z, l_c)$  can be obtained by using the recurrent expression:

$$U_{cj}(z,l_c) = U_{cj-1}(z,l_c) \bigotimes_{+} \left[ r_{cm_j} z^1 + \left( 1 - r_{cm_j} \right) z^0 \right].$$
(29)

According to its definition,  $p_{cj}(l_c)$  is the probability that the group of first j software versions produces  $k_c$  correct outputs and the group of first j-1 software versions produces  $k_c - 1$  correct outputs given that  $l_c$  software versions can be executed simultaneously. The coefficient  $\pi_{jk_c}$  in polynomial  $U_{cj}(z, l_c)$  is equal to the conditional probability that the group of first j software versions produces  $k_c$  correct outputs given that  $l_c$  software versions can be executed simultaneously.

In order to let the coefficient  $\pi_{jk_c}$  in polynomial  $U_{cj}(z, l_c)$  be equal to  $p_{cj}(l_c)$ , the term with the exponent equal to  $k_c$  should be removed from  $U_{cj-1}(z, l_c)$  before applying (29) (excluding the combination in which j - 1 first software versions produce  $k_c$  correct outputs while the  $m_j$ th software version fails).

If after the execution of *j* first software versions the number of correct outputs produced is *k* and  $k + n_c - j < k_c$ , then the required number of correct outputs  $k_c$  cannot be obtained even if all the  $n_c - j$  subsequent software versions produce correct outputs. Therefore, the terms  $\pi_{jk}z^k$  with  $k < k_c - n_c + j$  can be removed from  $U_{cj}(z, l_c)$ .

The above considerations lie at the base of the following algorithm for determining all of the probabilities  $p_{cj}(l_c)$  ( $k_c \le j \le n_c$ ).

- For the given *l<sub>c</sub>*, determine the order of software version termination *m<sub>1</sub>, m<sub>2</sub>,..., m<sub>n<sub>c</sub></sub>* using the algorithm from Section 4.1.2.
- Determine the *u*-function of each software version of cloud service *c* according to (26).
- (3) Define  $U_{c0}(z, l_c) = 1$ . For  $j = 1, 2, ..., n_c$ ,
  - (a) obtain U<sub>cj</sub>(z, l<sub>c</sub>) using (28) and, after collecting like terms, represent it in the form (29),
  - (b) remove from  $U_{cj}(z, l_c)$  all the terms  $\pi_{jk} z^k$  for which  $k < k_c n_c + j$ ,
  - (c) If  $j \ge k_c$ , assign  $p_{cj}(l_c) = \pi_{jk_c}$  and remove term  $\pi_{jk_c} z^{k_c}$  from  $U_{cj}(z, l_c)$ .

5.2. Evaluating Response Time Distribution of the Cloud Services Composition Application. Having the pairs  $p_{cj}(l_c(x))$ ,  $t_{cm_j}(l_c(x))$  for each possible realization  $l_c(x)$  of  $L_c$   $(1 \le x \le h_c)$  and probabilities  $\Pr\{L_c = l_c(x)\} = Q_c(x)$ , one can obtain the p.m.f. of random response times  $T_c$  for each cloud service

by applying (22). If the conditional p.m.f.  $p_{cj}(l_c(x))$ ,  $t_{cm}(l_c(x))$  are represented by the *u*-function

$$\widetilde{u}_{c}\left(z,l_{c}\left(x\right)\right) = \sum_{j=k_{c}}^{n_{c}} p_{cj}\left(l_{c}\left(x\right)\right) z^{t_{cm_{j}}\left(l_{c}\left(x\right)\right)}$$
(30)

then the *u*-function representing the p.m.f. of the random value  $T_c$  takes the form

$$\widetilde{U}_{c}(z) = \sum_{x=1}^{h_{c}} Q_{c}(x) \, \widetilde{u}_{c}(z, l_{c}(x)) \,. \tag{31}$$

In accordance with the four kinds of execution patterns of cloud services in cloud services composition application, we present four kinds of composition operators for *u*-function operation corresponding to formula (25):

 The composition operator ⊗<sub>sequ</sub> for sequence execution pattern:

$$\begin{split} \widetilde{U}(z) &= \widetilde{U}_{1}(z) \bigotimes_{\text{sequ}} \widetilde{U}_{2}(z) \\ &= \sum_{l_{1}=1}^{L_{1}} \Pr\left(T_{1l_{1}} = \widetilde{T}_{1l_{1}}\right) z^{\widetilde{T}_{1l_{1}}} \bigotimes_{\text{sequ}} \sum_{l_{2}=1}^{L_{2}} \Pr\left(T_{2l_{2}} = \widetilde{T}_{2l_{2}}\right) z^{\widetilde{T}_{2l_{2}}} \\ &= \sum_{l_{1}=1}^{L_{1}} \sum_{l_{2}=1}^{L_{2}} \Pr\left(T_{1l_{1}} = \widetilde{T}_{1l_{1}}\right) \Pr\left(T_{2l_{2}} = \widetilde{T}_{2l_{2}}\right) z^{\widetilde{T}_{1l_{1}} + \widetilde{T}_{2l_{2}}}. \end{split}$$
(32)

(2) The composition operator ⊗<sub>para</sub> for parallel execution pattern:

$$= \widetilde{U}_{1}(z) \bigotimes_{\text{para}} \widetilde{U}_{2}(z)$$

$$= \sum_{l_{1}=1}^{L_{1}} \Pr\left(T_{1l_{1}} = \widetilde{T}_{1l_{1}}\right) z^{\widetilde{T}_{1l_{1}}} \bigotimes_{\text{para}} \sum_{l_{2}=1}^{L_{2}} \Pr\left(T_{2l_{2}} = \widetilde{T}_{2l_{2}}\right) z^{\widetilde{T}_{2l_{2}}}$$

$$= \sum_{l_{1}=1}^{L_{1}} \sum_{l_{2}=1}^{L_{2}} \Pr\left(T_{1l_{1}} = \widetilde{T}_{1l_{1}}\right) \Pr\left(T_{2l_{2}} = \widetilde{T}_{2l_{2}}\right) z^{\max(\widetilde{T}_{1l_{1}}, \widetilde{T}_{2l_{2}})}.$$
(33)

(3) The composition operator ⊗<sub>split</sub> for split execution pattern:

$$\begin{split} \widetilde{U}(z) \\ &= \widetilde{U}_{1}(z) \bigotimes_{\text{split}} \widetilde{U}_{2}(z) \\ &= \sum_{l_{1}=1}^{L_{1}} \Pr\left(T_{1l_{1}} = \widetilde{T}_{1l_{1}}\right) z^{\widetilde{T}_{1l_{1}}} \bigotimes_{\text{split}} \sum_{l_{2}=1}^{L_{2}} \Pr\left(T_{2l_{2}} = \widetilde{T}_{2l_{2}}\right) z^{\widetilde{T}_{2l_{2}}} \\ &= p_{1} \sum_{l_{1}=1}^{L_{1}} \Pr\left(T_{1l_{1}} = \widetilde{T}_{1l_{1}}\right) z^{\widetilde{T}_{1l_{1}}} + p_{2} \sum_{l_{2}=1}^{L_{2}} \Pr\left(T_{2l_{2}} = \widetilde{T}_{2l_{2}}\right) z^{\widetilde{T}_{2l_{2}}}, \end{split}$$
(34)

where the  $p_1$  and  $p_2$  are the probabilities that the splits, corresponding to  $\widetilde{U}_1(z)$  and  $\widetilde{U}_2(z)$ , are chosen to execute.

(4) The composition operator ⊗<sub>loop</sub> for loop execution pattern.

The composition operator  $\otimes_{loop}$  can be expressed by multiple composition operators  $\otimes_{sequ}$ , because the loop execution pattern can be transformed to an accumulation of multiple sequence execution patterns. The number of composition operators  $\otimes_{sequ}$  transformed is equal to the number of times of repeated execution in loop pattern.

Hence, one can obtain the *u*-function  $\widetilde{U}(z)$  representing the p.m.f. of the random entire application response time *T* as

$$\widetilde{U}(z) = \bigotimes_{f} \left( \widetilde{U}_{1}(z), \dots, \widetilde{U}_{C}(z) \right)$$

$$= \bigotimes_{f} \left( \sum_{x=1}^{h_{c}} Q_{c}(x) \widetilde{u}_{c}(z, l_{c}(x)) \right),$$
(35)

where the composition operator  $\otimes_f$  is an abstract composition operator that it can be one of the composition operators  $\otimes_{sequ}, \otimes_{para}, \otimes_{split}, and \otimes_{loop}$ .

In accordance with the composition patterns in cloud services composition application, the concrete *u*-function  $\widetilde{U}(z)$  representing the p.m.f. of *T* can be obtained by replacing  $\otimes_f$  by one of the composition operators  $\otimes_{\text{sequ}}, \otimes_{\text{para}}, \otimes_{\text{split}}$ , and  $\otimes_{\text{loop}}$ .

5.3. Evaluating Response Time Distribution of Different Cloud Services Executed on the Same Hardware. Now consider the case where all of the software cloud services are consecutively executed on the same hardware consisting of h parallel identical modules with the availability a. The number of available parallel hardware modules H is random with p.m.f.  $Q(x) = Pr{H = x}, 1 \le x \le h$ , defined in the same way as in (17).

When H = x, the number of software versions that can be executed simultaneously in each cloud service c is  $l_c(x)$ . The *u*-functions representing the p.m.f. of the corresponding cloud service response time  $T_c$  are  $\tilde{u}_c(z, l_c(x))$  defined by (30). The *u*-function  $\hat{U}(z, x)$  representing the conditional p.m.f. of the entire application response time T (given that the number of available hardware modules is x) can be obtained for any x ( $1 \le x \le h$ ) as

$$\widehat{U}(z,x) = \bigotimes_{+} \left( \widetilde{u}_{1}(z,l_{1}(x)), \dots, \widetilde{u}_{C}(z,l_{C}(x)) \right)$$

$$= \prod_{c=1}^{C} \widetilde{u}_{c}(z,l_{c}(x)).$$
(36)

Having the p.m.f. of the random value *H* we obtain the *u*-function  $\widetilde{U}(z)$  representing the p.m.f. of *T* as

$$\widetilde{U}(z) = \sum_{x=1}^{H} Q(x) \widehat{U}(z, x).$$
(37)

 $\widetilde{U}(z)$ 

5.4. Optimizing the Structure of Cloud Service Composition Application with Fault-Tolerant Mechanism Based on UGF and GA. When a fault-tolerant cloud service application is designed, one has to choose software versions for each cloud service and find the sequence of their execution in order to achieve the entire application's greatest reliability subject to cost constraints. The software versions are chosen from a list of the available products. Each software version is characterized by its reliability, response time, and cost. The total cost of the entire application is defined according to the cost of its software versions. The cost for each software version can be the purchase cost if the software versions are commercial and the off-the-shelf cost, or it can be an estimate based upon the software version's size, complexity, and performance.

Assume that  $B_c$  functionally equivalent software versions are available for each cloud service c and that the number  $k_c$  of the software versions that should agree in each cloud service is predetermined. The choice of the software versions and the sequence of their execution in each cloud service determine the entire application's reliability and performance.

The permutation  $\mathbf{x}_c^*$  of  $B_c$  different integer numbers ranging from 1 to  $B_c$  determines the order of the software version that can be used in cloud service *c*. Let  $y_{cb} = 1$  if the software version *b* is chosen to be included in cloud service *c* and  $y_{cb} = 0$  otherwise. The binary vector  $y_c = \{y_{c1}, \ldots, y_{cB_c}\}$ determines the subset of software versions chosen for cloud service *c*. Having the vectors  $\mathbf{x}_c^*$  and  $\mathbf{y}_c$  one can determine the execution order  $x_c$  of the software versions chosen by removing from  $\mathbf{x}_c^*$  any number *b* for which  $\mathbf{y}_{cb} = 0$ . The total number of software versions in cloud service *c* (equal to the length of vector  $\mathbf{y}_c$  after removing the unchosen software versions) is determined as

$$n_c = \sum_{b=1}^{B_c} y_{cb}.$$
 (38)

The application structure optimization problem can now be formulated by finding vectors  $x_c$  for  $1 \le c \le C$  that maximize R(w) subject to cost constraint

$$\Omega = \sum_{c=1}^{C} \sum_{b \in x_c} \omega_{cb} \le \Omega^*,$$
(39)

where  $\omega_{cb}$  is the cost of software version *b* used in cloud service *c*,  $\Omega$  is the entire application cost and  $\Omega^*$  is the maximal allowable application cost. Note that the length of vectors  $\mathbf{x}_c$  can vary depending on the number of software versions chosen.

In order to encode the variable-length vectors  $\mathbf{x}_c$  in the GA using the constant length integer strings one can use  $(B_c + 1)$ -length strings containing permutations of numbers  $1, \ldots, B_c, B_c + 1$ . The numbers that appear before  $B_c + 1$  determine the vector  $\mathbf{x}_c$ . For example, for  $B_c = 5$  the permutations (2, 3, 6, 5, 1, 4) and (3, 1, 5, 4, 2, 6) correspond to  $\mathbf{x}_c = (2, 3)$  and  $\mathbf{x}_c = (3, 1, 5, 4, 2)$  respectively. Any possible vector  $\mathbf{x}_c$  can be represented by the corresponding integer substring containing the permutation of  $B_c + 1$  numbers. By combining *C* substrings corresponding to different cloud services one obtains the integer string *a*, that encodes the entire application structure.

The encoding method is used in which the single permutation defines the sequences of the software versions chosen in each of the *C* cloud services. The solution encoding string is a permutation of  $n = \sum_{c=1}^{C} (B_c + 1)$  integer numbers ranging from 1 to *n*. Each number *j* belonging to the interval  $\sum_{c=1}^{m-1} (B_c + 1) + 1 \le j \le \sum_{c=1}^{m} (B_c + 1)$  corresponds to software version  $j - \sum_{c=1}^{m-1} (B_c + 1)$  of cloud service. The relative order in which the numbers corresponding to the software versions of the same cloud service appear in the string determines the structure of this cloud service.

 TABLE 1: Parameters of fault-tolerant cloud services and software versions.

No. of cloud corrigon	$L_c$	k <sub>c</sub>	Software versions								
No. of cloud services				1	2	3	4	5	6	7	8
			с	5	15	7	8	12	6	_	_
1	1	1	τ	17	10	20	32	30	75	_	_
			r	0.71	0.85	0.85	0.89	0.95	0.98	_	_
			с	5	15	7	8	12	_	_	_
2	2	2	τ	28	55	35	55	58	_	_	_
			r	0.71	0.85	0.85	0.89	0.95	_	_	_
			с	4	3	4	6	5	4	9	6
3	4	3	τ	17	20	38	38	48	50	41	63
			r	0.80	0.80	0.86	0.90	0.90	0.94	0.98	0.98
			С	12	16	17	17	_	_	_	_
4	1	2	τ	17	10	20	32	_	_	_	_
			r	0.75	0.85	0.93	0.97	_	_	_	_
			с	5	9	11	7	12	_	_	_
5	3	1	τ	30	54	40	65	70	_	_	_
			r	0.70	0.80	0.80	0.80	0.89	_	_	_

Journal of Applied Mathematics



FIGURE 1: R(w) functions for the solutions obtained.

TABLE 2: Parameters of solutions obtained for w = 250.

$\Omega^*$	Sequence of software versions	$T_{\min}$	$T_{\rm max}$	Ω	R(250)	$\widetilde{\epsilon}(\infty)$
160	231   541   37162   324   214	166	307	159	0.913	188.34
140	34   241   64231   234   123	173	301	140	0.868	194.43
120	5   431   31562   43   21	205	249	119	0.752	217.07
100	3   241   4562   43   41	205	270	100	0.598	220.52

### 6. Illustrative Examples

Consider a fault-tolerant cloud services composition application consisting of five cloud services in serial running on fully available hardware. The parameters of the software versions that can be used in these cloud services are presented in Table 1. This table contains the values of  $k_c$  and  $L_c$  for each cloud service and the cost, reliability, and response time for each software version.

Two sets of solutions were obtained for the maximal allowable application response times w = 250 and w = 300. For each value of w, four different solutions were obtained for different cost constraints. These solutions are presented in Tables 2 and 3. The tables contain the application corresponding cost and reliability for each optimal solution, the expected conditional response time, minimal and maximal possible application response times, and the corresponding optimal execution sequences of the software versions chosen.

Comparing the entire application cost and the reliability of the optimal solutions corresponding to w = 250 and w = 300 in Tables 2 and 3, it can be seen that the entire application cost and the reliability of the optimal solution corresponding to w = 300 are always equal or greater than ones corresponding to w = 250 in the case of the same value of  $\Omega^*$ . Comparing the entire application cost and the reliability of the optimal solutions corresponding to the different 4 maximal allowable application costs in Tables 2 and 3, it can be seen that the entire application cost and the reliability of the optimal solution corresponding to larger  $\Omega^*$  are always equal or greater than ones corresponding to smaller  $\Omega^*$  in the case of the same value of w.

From Tables 2 and 3, it can also be seen that the software versions executed in practice gradually become more and more along with the growth of the value of  $\Omega^*$ .

These phenomenon above indicates that the selection of suitable  $\Omega^*$  and w is helpful to improve the reliability of the cloud services composition application and cut down the cost.

To help the designers of the cloud services composition application to select the suitable  $\Omega^*$  and w, the values of the functions R(w) of all of solutions obtained are drawn in Figure 1. At first, the designers can intuitively find out which curves cross over the value of reliability demand. On this basis, the designers can easily find which points (solution) can meet the maximal allowable response time. The approach abovementioned can be easily realized by software. Thus, it can be applied in online prediction and optimization situation.

$\Omega^*$	Sequence of software versions	$T_{\min}$	$T_{\rm max}$	Ω	R(300)	$\widetilde{\epsilon}(\infty)$
160	341   4521   85632   324   41	188	369	160	0.951	210.82
140	53   541   28361   431   51	173	301	140	0.868	194.43
120	6   241   61372   241   31	240	307	120	0.813	252.87
100	4   142   2386   43   41	219	295	100	0.672	238.05

TABLE 3: Parameters of solutions obtained for w = 300.

### 7. Conclusions

The traditional reliability and performance prediction and optimization techniques, for example, Markov model and state space analysis, have some defects such as being too time consuming and easy to cause state space explosion and unsatisfied the assumptions of component execution independence by Markov model. Aiming at the defects of Markov model, an optimization model of reliability and performance based on MSS for cloud services application is proposed in this paper, which eliminates the limitation for component execution independence, and more fits the actual execution of cloud services composition application. On this basis, aiming at the defects of state space analysis technique, a fast optimization algorithm with very small time consumption based on UGF and GA for the reliability and performance of cloud services composition application is presented in this paper, which eliminates the risk of state space explosion. The model and algorithm presented in this paper can be applied in online prediction and optimization for reliability and performance of cloud services composition application.

### Acknowledgment

This research was supported by the National Natural Science Funds Fund of China (61172084); Science and Technology Support Program of Hubei Province of China (2013BHE022); Natural Science Foundation of Hubei Province of China (2013CFC026); Key new product research and development of Hubei Province of China (2012BBA25002, 2012IHA015).

### References

- W. Venters and E. A. Whitley, "A critical review of cloud computing: researching desires and realities," *Journal of Information Technology*, vol. 27, no. 3, pp. 179–197, 2012.
- [2] L. Qi, W. Dou, X. Zhang, and J. Chen, "A QoS-aware composition method supporting cross-platform service invocation in cloud environment," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1316–1329, 2012.
- [3] J. Leukel, S. Kirn, and T. Schlegel, "Supply chain as a service: a cloud perspective on supply chain systems," *IEEE Systems Journal*, vol. 5, no. 1, pp. 16–27, 2011.
- [4] C. Y. Ming and P. Y. Jen, "A QoS aware services mashup model for cloud computing applications," *Journal of Industrial Engineering and Management*, vol. 5, no. 2, pp. 457–472, 2012.
- [5] L. Zhao, Y. Ren, M. Li, and K. Sakurai, "Flexible service selection with user-specific QoS support in service-oriented architecture," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 962–973, 2012.

- [6] W. Dou, L. Qi, X. Zhang, and J. Chen, "An evaluation method of outsourcing services for developing an elastic cloud platform," *Journal of Supercomputing*, vol. 63, no. 1, pp. 1–23, 2010.
- [7] Q. Wu, Z. B. Li, Y. Y. Yin, and H. Zeng, "Adaptive service selection method in mobile cloud computing," *China Communications*, vol. 9, no. 12, pp. 46–55, 2012.
- [8] S.-Y. Hwang, E.-P. Lim, C.-H. Lee, and C.-H. Chen, "Dynamic web service selection for reliable web service composition," *IEEE Transactions on Services Computing*, vol. 1, no. 2, pp. 104– 116, 2008.
- [9] Y.-K. Lin and P.-C. Chang, "Reliability evaluation of a computer network in cloud computing environment subject to maintenance budget," *Applied Mathematics and Computation*, vol. 219, no. 8, pp. 3893–3902, 2012.
- [10] G. Katsaros, G. Kousiouris, S. V. Gogouvitis, D. Kyriazis, A. Menychtas, and T. Varvarigou, "A self-adaptive hierarchical monitoring mechanism for clouds," *Journal of Systems and Software*, vol. 85, no. 5, pp. 1029–1041, 2012.
- [11] M. L. Chiang, "Efficient diagnosis protocol to enhance the reliability of a cloud computing environment," *Journal of Network* and Systems Management, vol. 20, no. 4, pp. 579–600, 2012.
- [12] X. Yang and H. Zhang, "Cloud computing and SOA convergence research," in *Proceedings of the 5th International Symposium on Computational Intelligence and Design (ISCID '12)*, vol. 1, pp. 330–335, Hangzhou, China, 2012.
- [13] R. Sharma, M. Sood, and D. Sharma, "Modeling Cloud SaaS with SOA and MDA," in *Proceedings of the 1st International Conference on Advances in Computing and Communications*, vol. 190, pp. 511–518, Kochi, India, July 2011.
- [14] R. K. Das, S. Patnaik, and A. K. Misro, "Adoption of cloud computing in e-governance," in *Proceedings of the 1st International Conference on Computer Science and Information Technology*, vol. 133 of *Communications in Computer and Information Science*, pp. 161–172, January 2011.
- [15] L. Qi, W. Dou, X. Zhang, and J. Chen, "A QoS-aware composition method supporting cross-platform service invocation in cloud environment," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1316–1329, 2012.
- [16] X. Chen, X. Liu, F. Fang, X. Zhang, and G. Huang, "Management as a service: an empirical case study in the internetware cloud," in *Proceedings of the IEEE International Conference on E-Business Engineering (ICEBE '10)*, pp. 470–473, Shanghai, China, November 2010.
- [17] P. Kumar, V. K. Sehgal, D. S. Chauhan, P. K. Gupta, and M. Diwakar, "Effective ways of secure, private and trusted cloud computing," *International Journal of Computer Science Issues*, vol. 8, no. 3, pp. 412–420, 2011.
- [18] X. Xu, C. Cheng, and J. Xiong, "Reliable integrated model of cloud & client computing based on multi-agent," *Journal of Computational Information Systems*, vol. 6, no. 14, pp. 4767– 4774, 2010.
- [19] J. Liu, J. Zhou, J. Wang, X. Chen, and H. Zhou, "Cloud service: automatic construction and evolution of software process

problem-solving resource space," *Journal of Supercomputing*, pp. 1–25, 2010.

- [20] J. Murhland, "Fundamental concepts and relations for reliability analysis of multistate system," in *Reliability and Fault Tree Analysis. Theoretical and Applied Aspects of System Reliability*, pp. 581–618, SIAM, 1975.
- [21] E. El-Neweihi, F. Proschan, and J. Sethuraman, "Multistate coherent systems," *Journal of Applied Probability*, vol. 15, no. 4, pp. 675–688, 1978.
- [22] R. E. Barlow and A. S. Wu, "Coherent systems with multistate components," *Mathematics of Operations Research*, vol. 3, no. 4, pp. 275–281, 1978.
- [23] S. Ross, "Mulivalued state element system," *Annals of Probability*, no. 7, pp. 379–383, 1979.
- [24] A. Lisnianski and G. Levitin, Multi-State System Reliability: Assessment, Optimization and Applications, vol. 6 of Series on Quality, Reliability & Engineering Statistics, World Scientific, River Edge, NJ, USA, 2003.
- [25] I. Ushakov, "Optimal standby problems and a universal generating function," *Soviet Journal of Computer Systems Science*, vol. 25, pp. 79–82, 1987.

## *Research Article*

## **Reflected Backward Stochastic Differential Equations Driven by Countable Brownian Motions**

## Pengju Duan,<sup>1,2</sup> Min Ren,<sup>2</sup> and Shilong Fei<sup>2</sup>

<sup>1</sup> Laboratory of Intelligent Information Processing, Suzhou University, Anhui 234000, China <sup>2</sup> Department of Mathematics, Suzhou University, Anhui 234000, China

Correspondence should be addressed to Pengju Duan; pjduan1981@hotmail.com

Received 16 June 2013; Accepted 15 September 2013

Academic Editor: Lotfollah Najjar

Copyright © 2013 Pengju Duan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper deals with a new class of reflected backward stochastic differential equations driven by countable Brownian motions. The existence and uniqueness of the RBSDEs are obtained via Snell envelope and fixed point theorem.

### 1. Introduction

The nonlinear backward stochastic differential equations (BSDEs in short) were introduced by Pardoux and Peng [1], who proved the existence and uniqueness of the solution under the Lipschitz conditions for giving the probabilistic interpretation of semilinear parabolic partial differential equations. Since then, many authors were devoted to studying the BSDEs (see, e.g., [2–8] and the references therein). At present, the theory of BSDEs becomes a powerful tool to solve practical matters. In 1994, Pardoux and Peng [9] firstly studied the backward doubly stochastic differential equations (BDSDEs in short), which are driven by two kinds of Brownian motions. Later, Boufoussi et al. [10] established the connection between a class of generalized BDSDEs and semilinear stochastic partial differential equations with a Neumann boundary condition.

Reflected backward differential equations (RBSDEs in short) were introduced by El Karoui et al. [11]. Later, many researchers discussed various kinds of RBSDEs for their deep application in mathematical finance and partial differential equations. Ren and Hu [12] proposed the RBSDEs, driven by Teugels martingales and Brownian motion, and derived the existence and uniqueness of the solution by means of the Snell envelope and the fixed point theorem when the barrier was right continuous with left limits. Ren and El Otmani [13] discussed the generalized reflected BSDEs driven by Lévy process. Recently, Ren et al. [14] studied a new class of reflected backward doubly stochastic differential equations driven by Lévy process and Brownian motion.

As in all the previous works, the equations are driven by finite Brownian motions. To the best of our knowledge, there are no papers on the reflected backward stochastic differential equations driven by countable Brownian motions. In this paper, we aim to derive the existence and uniqueness of the solution for the RBSDEs driven by countable Brownian motions.

The structure of the paper is organized as follows. In Section 2, we give some notations. Section 3 is devoted to the main result.

### 2. Notations

Let *T* be a positive constant. Throughout the paper  $(\Omega, \mathcal{F}, \mathbb{P})$  is a complete probability space equipped with the natural filtration  $\{\mathcal{F}_t\}_{t\geq 0}$  satisfying the usual conditions.  $\{\beta_j(t)\}_{j=1}^{\infty}$  are mutual independent one-dimensional standard Brownian motions on the probability space. W(t) is a standard Brownian motion on  $\mathbb{R}^d$  which is independent of  $\beta_i(t)$ . Assume that

$$\mathscr{F}_{t} = \left(\bigvee_{j=1}^{\infty} \mathscr{F}_{t,T}^{\beta_{j}}\right) \bigvee \mathscr{F}_{t}^{W} \bigvee \mathscr{N}, \tag{1}$$

where for any process  $\{\eta_t\}$ ,  $\mathcal{F}_{s,t}^{\eta} = \sigma\{\eta_r - \eta_s : s \le r \le t\}$ ,  $\mathcal{F}_t^{\eta} = \mathcal{F}_{0,t}^{\eta}$ , and  $\mathcal{N}$  denotes the class of  $\mathbb{P}$ -null sets of  $\mathcal{F}$ .

For the convenience, let us introduce some spaces:

- (i)  $\mathscr{H}^2 = \{(\varphi_t)_{0 \le t \le T}: \text{ an } \mathscr{F}_t \text{-progressively measurable, } \mathbb{R} \text{-valued process such that } E \int_0^T |\varphi_t|^2 dt < \infty \};$
- (ii)  $S^2 = \{(\psi_t)_{0 \le t \le T}: \text{ an } \mathcal{F}_t\text{-progressively measurable,} \mathbb{R}^d\text{-valued continuous process such that} E(\sup_{0 \le t \le T} |\psi_t|^2) < \infty\};$
- (iii)  $\mathscr{A}^2 = \{(K_t)_{0 \le t \le T}: \text{ an } \mathscr{F}_t \text{ -adapted, continuous, increasing process such that } K_0 = 0, E|K_t|^2 < \infty\}.$

With the previous preparations, we consider the following RBSDEs:

$$Y_{t} = \xi + \int_{t}^{T} f\left(s, Y_{s}, Z_{s}\right) ds$$
  
+ 
$$\sum_{j=1}^{\infty} \int_{t}^{T} g_{j}\left(s, Y_{s}, Z_{s}\right) d\beta_{j}\left(s\right)$$
  
- 
$$\int_{t}^{T} Z_{s} dW\left(s\right) + K_{T} - K_{t}, \quad 0 \le t \le T,$$
(2)

where  $f: \Omega \times [0,T] \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$  and  $g_j: \Omega \times [0,T] \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ .

*Definition 1.* A solution of (2) is a triple of  $\mathbb{R} \times \mathbb{R}^d \times \mathbb{R}_+$  value process  $(Y_t, Z_t, K_t)_{0 \le t \le T}$ , which satisfies (2), and

- $\begin{array}{l} (\mathrm{i}) \ (Y_t, Z_t, K_t)_{0 \leq t \leq T} \in \, \mathcal{S}^2 \times \mathcal{H}^2 \times \mathcal{A}^2; \\ (\mathrm{ii}) \ Y_t \geq S_t; \end{array}$
- (iii)  $K_t$  is a continuous and increasing process with  $K_0 = 0$ and  $\int_0^T (Y_t - S_t) dK_t = 0$ .

In order to get the solution of (2), we propose the following assumptions:

- (H1)  $\xi$  is an  $\mathscr{F}_T$  measurable square integrable random variable;
- (H2) the obstacle { $S_t : 0 \le t \le T$ } is an  $\mathcal{F}_t$ -progressive measurable continuous real valued process which satisfies  $E \sup_{0 \le t \le T} (S_t)^2 < \infty$ . We always assume that  $S_T \le \xi$ , a.s.;
- (H3)  $f(\cdot, y, z)$  and  $g_j(\cdot, y, z)$  are two progressive measurable functions such that, for any  $t \in [0, T]$ ,  $y_1, y_2 \in \mathbb{R}$ ,  $z_1, z_2 \in \mathbb{R}^d$ ,

(3a) 
$$f(s, \cdot, \cdot)$$
 is continuous and  $|f(s, y, z)| \le M(1 + |y| + |z|)$ ;

(3b) 
$$E \int_0^T |f(t,0,0)|^2 dt < \infty$$
,  
 $\sum_{j=1}^\infty E \int_0^T |g_j(t,0,0)|^2 dt < \infty$ ;

(3c) 
$$|f(s, y_1, z_1) - f(s, y_2, z_2)|^2 \leq C(|y_1 - y_2|^2 + |z_1 - z_2|^2), \quad |g_j(s, y_1, z_1) - g_j(s, y_2, z_2)|^2 \leq C_j |y_1 - y_2|^2 + \alpha_j |z_1 - z_2|^2, \text{ where } M, C, C_j, \text{ and } \alpha_j \text{ are nonnegative constants with } \sum_{j=1}^{\infty} C_j < \infty \text{ and } \alpha = \sum_{j=1}^{\infty} \alpha_j < 1.$$

### 3. Main Result

In order to get the solution of (2), we consider the following RBSDEs driven by finite Brownian motions:

$$Y_{t} = \xi + \int_{t}^{T} f(s, Y_{s}, Z_{s}) ds$$
  
+  $\sum_{j=1}^{n} \int_{t}^{T} g_{j}(s, Y_{s}, Z_{s}) d\beta_{j}(s)$  (3)  
-  $\int_{t}^{T} Z_{s} dW(s) + K_{T} - K_{t}, \quad 0 \le t \le T.$ 

Firstly, we consider a special case of (3); that is, the functions f and g do not depend on (Y, Z):

$$Y_{t} = \xi + \int_{t}^{T} f(s) ds + \sum_{j=1}^{n} \int_{t}^{T} g_{j}(s) d\beta_{j}(s) - \int_{t}^{T} Z_{s} dW(s)$$
(4)  
+  $K_{T} - K_{t}, \quad 0 \le t \le T, \ n \ge 1.$ 

We will get the existence and uniqueness of the solution of (4) by means of Snell envelope and martingale representation theorem.

**Theorem 2.** Assume that (H1)-(H2),  $f \in \mathcal{H}^2$ ,  $g \in \mathcal{H}^2$ . Then, there exists a triple  $(Y_t, Z_t, K_t)_{0 \le t \le T} \in \mathcal{S}^2 \times \mathcal{H}^2 \times \mathcal{A}^2$  which is a solution of (4).

Proof. Let

$$\mathscr{C}_{t} = \mathscr{F}_{t}^{W} \bigvee \left( \bigvee_{j=1}^{n} \mathscr{F}_{t,T}^{\beta_{j}} \right), \tag{5}$$

and we define  $\eta = {\eta_t}_{0 \le t \le T}$  as

$$\eta_{t} = \xi \mathbf{1}_{\{t=T\}} + S_{t} \mathbf{1}_{\{t
(6)$$

Then,  $\eta$  is  $\mathcal{C}_t$ -adapted continuous process; furthermore;

$$\sup_{0 \le t \le T} \left| \eta_t \right| \in L^2(\Omega) \,. \tag{7}$$

So, the Snell envelope of  $\eta$  is given by

$$S_{t}(\eta) = \operatorname{ess\,sup}_{\nu \in \mathcal{T}} E\left[\eta_{\nu} \mid \mathscr{C}_{t}\right], \tag{8}$$

where  $\mathcal{T}$  is the set of all  $\mathcal{C}_t$  stopping time such that  $0 \le \nu \le T$ .

By the definition of  $\eta$ , we can deduce that

$$E\left[\sup_{0\leq t\leq T}\left|S_{t}\left(\eta\right)\right|^{2}\right]<\infty.$$
(9)

Due to the Doob-Meyer decomposition, we have

$$S_{t}(\eta) = E\left[\xi + \int_{0}^{T} f(s) ds + \sum_{j=1}^{n} \int_{0}^{T} g_{j}(s) d\beta_{j}(s) + K_{T} \mid \mathscr{C}_{t}\right] - K_{t},$$
(10)

where  $\{K_t\}_{0 \le t \le T}$  is a  $\mathscr{C}_t$ -adapted, continuous, and nondecreasing process such that  $K_0 = 0$  and  $EK_T^2 < \infty$ . So, we have

$$E\left[\sup_{0\leq t\leq T}\left|E\left[\xi+\int_{0}^{T}f\left(s\right)ds\right.\right.\right.\right.\right.\right.\right.$$
$$\left.+\sum_{j=1}^{n}\int_{0}^{T}g_{j}(s)d\beta_{j}(s)+K_{T}\left|\mathscr{C}_{t}\right]\right|^{2}\right]<\infty.$$
$$(11)$$

Martingale representation theorem yields that there exists  $\mathscr{C}_t$ -progressive measurable process  $\{Z_t\} \in \mathbb{R}^d$  such that

$$M_{t} \triangleq E\left[\xi + \int_{0}^{T} f(s) \,\mathrm{d}s + \sum_{j=1}^{n} \int_{0}^{T} g_{j}(s) \,\mathrm{d}\beta_{j}(s) + K_{T} \mid \mathscr{C}_{t}\right]$$
$$= M_{0} + \int_{0}^{t} Z_{s} \mathrm{d}W(s), \quad 0 \le t \le T.$$
(12)

Let  $Y_t = \operatorname{ess\,sup}_{\nu \in \mathcal{T}} E[\xi 1_{\{\nu = T\}} + S_{\nu} 1_{\{\nu < T\}} + \int_t^{\nu} f(s) ds + \sum_{j=1}^n \int_t^{\nu} g_j(s) d\beta_j(s) | \mathcal{C}_t]$ ; then,

$$Y_{t} + \int_{0}^{t} f(s) ds + \sum_{j=1}^{n} \int_{0}^{t} g_{j}(s) d\beta_{j}(s)$$
  
=  $S_{t}(\eta) = M_{t} - K_{t}$  (13)  
=  $M_{0} + \int_{0}^{t} Z_{s} dW(s) - K_{t}, \quad 0 \le t \le T.$ 

Therefore,

$$Y_{t} = \xi + \int_{t}^{T} f(s) ds + \sum_{j=1}^{n} \int_{t}^{T} g_{j}(s) d\beta_{j}(s) - \int_{t}^{T} Z_{s} dW(s) + K_{T} - K_{t}.$$
(14)

By the definitions of  $Y_t$  and  $S_t(\eta)$ ,  $\xi \ge S_T$ ,

$$Y_{t} + \int_{0}^{t} f(s) ds + \sum_{j=1}^{n} \int_{0}^{t} g_{j}(s) d\beta_{j}(s)$$

$$= S_{t}(\eta) \ge \eta_{t}$$

$$= \xi \mathbf{1}_{\{t=T\}} + S_{t} \mathbf{1}_{\{t

$$+ \sum_{j=1}^{n} \int_{0}^{t} g_{j}(s) d\beta_{j}(s)$$

$$\ge S_{T} \mathbf{1}_{\{t=T\}} + S_{t} \mathbf{1}_{\{t

$$+ \sum_{j=1}^{n} \int_{0}^{t} g_{j}(s) d\beta_{j}(s) .$$
(15)$$$$

So, we have  $Y_t \ge S_t$ .

Finally, from Hamadène [15], we get  $\int_0^T (S_t(\eta) - \eta_t) dK_t = 0$ ; that is,

$$\int_{0}^{T} (Y_t - S_t) \, \mathrm{d}K_t = 0. \tag{16}$$

It shows that the process  $(Y_t, Z_t, K_t)_{0 \le t \le T}$  is a solution of (4).

**Theorem 3.** Under the assumptions of (H1)–(H3), there exists a unique solution  $(Y_t, Z_t, K_t)_{0 \le t \le T}$  of (3).

*Proof.* Let  $\mathscr{P} = \mathscr{S}^2 \times \mathscr{H}^2$  be endowed with the norm

$$\|(Y,Z)\|_{\beta} = \left(E\left[\int_{0}^{T} e^{\beta s} \left(\left|Y_{s}\right|^{2} + \left|Z_{s}\right|^{2}\right) \mathrm{d}s\right]\right)^{1/2}$$
(17)

for a suitable constant  $\beta > 0$ . We define the map  $\Phi$  from  $\mathscr{P}$  into itself and  $(\tilde{Y}, \tilde{Z})$  and  $(\tilde{Y}', \tilde{Z}')$  are two elements of  $\mathscr{P}$ . Define  $(Y, Z) = \Phi(\tilde{Y}, \tilde{Z}), (Y', Z') = \Phi(\tilde{Y}', \tilde{Z}')$ , where (Y, Z, K) and (Y', Z', K') are solutions of (4)

associated with  $(\xi, f(t, \tilde{Y}, \tilde{Z}), g_j(t, \tilde{Y}, \tilde{Z}), S)$ , and  $(\xi, f(t, \tilde{Y'}, \tilde{Z'}), g_j(t, \tilde{Y'}, \tilde{Z'}), S')$ , respectively. Set  $(\overline{Y}, \overline{Z}) = (Y_t - Y'_t, Z_t - Z'_t)$  and

$$\Psi_{M}(x) = x^{2} \mathbf{1}_{\{-M \le x \le M\}} + M (2x - M) \mathbf{1}_{\{x > M\}} - M (2x + M) \mathbf{1}_{\{x < -M\}}.$$
(18)

If we define  $\Psi'_M(x)/x = 2$ , when x = 0, then,  $0 \le \Psi'_M(\overline{Y_s})/\overline{Y_s} \le 2$ . Applying Itô formula to  $e^{\beta t} \Psi_M(\overline{Y_s})$ , we have

$$e^{\beta t} \Psi_{M}\left(\overline{Y_{t}}\right) + \beta \int_{t}^{T} e^{\beta s} \Psi_{M}\left(\overline{Y_{s}}\right) ds + \int_{t}^{T} e^{\beta s} \mathbf{1}_{\{-M \leq \overline{Y_{s}} \leq M\}} \left|\overline{Z}_{s}\right|^{2} ds = \int_{t}^{T} e^{\beta s} \Psi_{M}'\left(\overline{Y_{s}}\right) \left(f\left(s, \overline{Y_{s}}, \overline{Z_{s}}\right) - f\left(s, \overline{Y_{s}'}, \overline{Z_{s}'}\right)\right) ds + \sum_{j=1}^{n} \int_{t}^{T} e^{\beta s} \mathbf{1}_{\{-M \leq \overline{Y_{s}} \leq M\}} \left|g_{j}\left(s, \overline{Y_{s}}, \overline{Z_{s}}\right) - g_{j}\left(s, \overline{Y_{s}'}, \overline{Z_{s}'}\right)\right|^{2} ds$$

$$- \sum_{j=1}^{n} \int_{t}^{T} e^{\beta s} \Psi_{M}'\left(\overline{Y_{s}}\right) \left(g_{j}\left(s, \overline{Y_{s}}, \overline{Z_{s}}\right) - g_{j}\left(s, \overline{Y_{s}'}, \overline{Z_{s}'}\right)\right) d\beta_{j}(s)$$

$$- g_{j}\left(s, \overline{Y_{s}'}, \overline{Z_{s}'}\right) d\beta_{j}(s)$$

$$= \int_{t}^{T} e^{\beta s} \Psi_{M}'(\overline{Y_{s}}) \overline{Z}_{s} dW(s)$$
$$+ \int_{t}^{T} e^{\beta s} \Psi_{M}'(\overline{Y_{s}}) \left( dK_{s} - dK_{s}' \right).$$

Taking expectation on both sides of (19) and noticing that  $\int_{t}^{T} e^{\beta s} \Psi'_{M}(\overline{Y_{s}}) (dK_{s} - dK'_{s}) \leq 0$ , we have

$$\begin{split} Ee^{\beta t}\Psi_{M}\left(\overline{Y_{t}}\right) + E\beta \int_{t}^{T} e^{\beta s}\Psi_{M}\left(\overline{Y_{s}}\right) \mathrm{d}s \\ &+ E\int_{t}^{T} e^{\beta s}\mathbf{1}_{\{-M \leq \overline{Y_{s}} \leq M\}} \left|\overline{Z}_{s}\right|^{2} \mathrm{d}s \\ &\leq E\int_{t}^{T} e^{\beta s}\Psi_{M}'\left(\overline{Y_{s}}\right)\left(f\left(s,\widetilde{Y_{s}},\widetilde{Z_{s}}\right) - f\left(s,\widetilde{Y_{s}'},\widetilde{Z_{s}'}\right)\right) \mathrm{d}s \\ &+ \sum_{j=1}^{n} E\int_{t}^{T} e^{\beta s}\mathbf{1}_{\{-M \leq \overline{Y_{s}} \leq M\}} \left|g_{j}\left(s,\widetilde{Y_{s}},\widetilde{Z_{s}}\right) - g_{j}\left(s,\widetilde{Y_{s}'},\widetilde{Z_{s}'}\right)\right|^{2} \mathrm{d}s \end{split}$$

$$\leq 2E \int_{t}^{T} e^{\beta s} \overline{Y_{s}} \left( f\left(s, \widetilde{Y_{s}}, \widetilde{Z_{s}}\right) - f\left(s, \widetilde{Y_{s}'}, \widetilde{Z_{s}'}\right) \right) ds$$

$$+ \sum_{j=1}^{n} E \int_{t}^{T} e^{\beta s} \left| g_{j}\left(s, \widetilde{Y_{s}}, \widetilde{Z_{s}}\right) - g_{j}\left(s, \widetilde{Y_{s}'}, \widetilde{Z_{s}'}\right) \right|^{2} ds$$

$$\leq \frac{2C}{1-\alpha} E \int_{t}^{T} e^{\beta s} \left| \overline{Y_{s}} \right|^{2} ds$$

$$+ \left( \sum_{j=1}^{\infty} C_{j} + \frac{1-\alpha}{2} \right) E \int_{t}^{T} e^{\beta s} \left| \widetilde{Y_{s}} - \widetilde{Y_{s}'} \right|^{2} ds$$

$$+ \frac{1+\alpha}{2} E \int_{t}^{T} e^{\beta s} \left| \widetilde{Z_{s}} - \widetilde{Z_{s}'} \right|^{2} ds.$$
(20)

Let  $\gamma = 2C/(1 - \alpha)$ ,  $\overline{C} = 2(\sum_{j=1}^{\infty} C_j + ((1 - \alpha)/2))/(1 + \alpha)$ ,  $\beta = \gamma + \overline{C}$ , and  $M \to \infty$ ; we have

$$\overline{C}E \int_{t}^{T} e^{\beta s} |Y_{s} - Y_{s}'|^{2} ds + E \int_{t}^{T} e^{\beta s} |Z_{s} - Z_{s}'|^{2} ds$$

$$\leq \frac{1 + \alpha}{2} E \int_{t}^{T} e^{\beta s} \left(\overline{C} |\widetilde{Y}_{s} - \widetilde{Y}_{s}'|^{2} + |\widetilde{Z}_{s} - \widetilde{Z}_{s}'|^{2}\right);$$
(21)

that is,

$$\|(Y_{s}, Z_{s})\|_{\beta}^{2} \leq \frac{1+\alpha}{2} \|(Y_{s}', Z_{s}')\|_{\beta}^{2}.$$
 (22)

It follows that  $\Phi$  is a strict contraction on  $\mathscr{P}$  with the norm  $\|\cdot\|_{\beta}$ , where  $\beta$  is defined as above. Then,  $\Phi$  has a fixed point (Y, Z, K) which is the unique solution of (4) from the Burkholder-Davis-Gundy inequality.

With all the preparations, we will give the main result of this paper as follows.

**Theorem 4.** Under the conditions of (H1)–(H3), there exists a unique solution  $(Y_t, Z_t, K_t)_{0 \le t \le T} \in S^2 \times \mathcal{H}^2 \times \mathcal{A}^2$  of (2).

*Proof (existence).* By Theorem 3, for any  $n \ge 1$ , there exists a unique solution of (3), denoted by  $(Y_t^n, Z_t^n, K_t^n)$ ,

$$Y_{t}^{n} = \xi + \int_{t}^{T} f(s, Y_{s}^{n}, Z_{s}^{n}) ds + \sum_{j=1}^{n} \int_{t}^{T} g_{j}(s, Y_{s}^{n}, Z_{s}^{n}) d\beta_{j}(s)$$
(23)  
$$- \int_{t}^{T} Z_{s}^{n} dW(s) + K_{T}^{n} - K_{t}^{n}.$$

In the following parts, we will claim that  $(Y_t^n, Z_t^n, K_t^n)$  is a Cauchy sequence in  $\mathcal{S}^2 \times \mathcal{H}^2 \times \mathcal{A}^2$ . Without loss of generality,

we let n < m. Applying general Itô formula to  $|Y_t^n - Y_t^m|^2$ , we have.

$$\begin{split} |Y_{t}^{n} - Y_{t}^{m}|^{2} + \int_{t}^{T} |Z_{s}^{n} - Z_{s}^{m}|^{2} ds \\ &= 2 \int_{t}^{T} (Y_{s}^{n} - Y_{s}^{m}) \left( f\left(s, Y_{s}^{n}, Z_{s}^{n}\right) - f\left(s, Y_{s}^{m}, Z_{s}^{m}\right) \right) ds \\ &+ \sum_{j=n+1}^{m} \int_{t}^{T} \left| g_{j}\left(s, Y_{s}^{n}, Z_{s}^{n}\right) - g_{j}\left(s, Y_{s}^{m}, Z_{s}^{m}\right) \right|^{2} ds \\ &- 2 \sum_{j=n+1}^{m} \int_{t}^{T} (Y_{s}^{n} - Y_{s}^{m}) \left( g_{j}\left(s, Y_{s}^{n}, Z_{s}^{n}\right) - g_{j}\left(s, Y_{s}^{m}, Z_{s}^{n}\right) \right) d\beta_{j}\left(s\right) \\ &- 2 \int_{t}^{T} \left(Y_{s}^{n} - Y_{s}^{m}\right) \left(Z_{s}^{n} - Z_{s}^{m}\right) dW\left(s\right) \\ &+ 2 \int_{t}^{T} \left(Y_{s}^{n} - Y_{s}^{m}\right) \left(dK_{s}^{n} - dK_{s}^{m}\right). \end{split}$$

$$(24)$$

Taking expectation on both sides of (24) and noting that  $\int_t^T (Y_s^n - Y_s^m) (\mathrm{d} K_s^n - \mathrm{d} K_s^m) \leq 0$ , we obtain

$$E|Y_{t}^{n} - Y_{t}^{m}|^{2} + E \int_{t}^{T} |Z_{s}^{n} - Z_{s}^{m}|^{2} ds$$

$$\leq 2E \int_{t}^{T} (Y_{s}^{n} - Y_{s}^{m}) (f(s, Y_{s}^{n}, Z_{s}^{n}) - f(s, Y_{s}^{m}, Z_{s}^{m})) ds$$

$$+ \sum_{j=n+1}^{m} E \int_{t}^{T} |g_{j}(s, Y_{s}^{n}, Z_{s}^{n}) - g_{j}(s, Y_{s}^{m}, Z_{s}^{m})|^{2} ds.$$
(25)

By (H3) and elementary inequality  $2ab \le \beta a^2 + (1/\beta)b^2$ ,  $\beta > 0$ , we obtain

$$\begin{split} E|Y_{t}^{n} - Y_{t}^{m}|^{2} + E \int_{t}^{T} |Z_{s}^{n} - Z_{s}^{m}|^{2} \mathrm{d}s \\ &\leq \frac{2C}{1 - \alpha} E \int_{t}^{T} |Y_{s}^{n} - Y_{s}^{m}|^{2} \mathrm{d}s + \frac{1 - \alpha}{2} E \int_{t}^{T} |Y_{s}^{n} - Y_{s}^{m}|^{2} \mathrm{d}s \\ &+ \frac{1 - \alpha}{2} E \int_{t}^{T} |Z_{s}^{n} - Z_{s}^{m}|^{2} \mathrm{d}s + \alpha E \int_{t}^{T} |Z_{s}^{n} - Z_{s}^{m}|^{2} \mathrm{d}s \\ &+ \left[\sum_{j=n+1}^{m} C_{j}\right] E \int_{t}^{T} |Y_{s}^{n} - Y_{s}^{m}|^{2} \mathrm{d}s. \end{split}$$

Furthermore,

$$E|Y_{t}^{n} - Y_{t}^{m}|^{2} + \frac{1 - \alpha}{2}E\int_{t}^{T}|Z_{s}^{n} - Z_{s}^{m}|^{2}ds$$
  
$$\leq C_{p}E\int_{t}^{T}|Y_{s}^{n} - Y_{s}^{m}|^{2}ds,$$
(27)

where  $C_p = (2C/(1 - \alpha)) + ((1 - \alpha)/2) + \sum_{j=n+1}^{m} C_j$ . By Gronwall's inequality and Burkholder-Davis-Gundy

By Gronwall's inequality and Burkholder-Davis-Gundy inequality, we have

$$E\left[\sup_{0\le t\le T}\int_t^T \left|Y_s^n-Y_s^m\right|^2 \mathrm{d}s\right] \longrightarrow 0.$$
 (28)

Denote the limit of  $(Y_t^n, Z_t^n, K_t^n)$  by  $(Y_t, Z_t, K_t)$ ; we will show that  $(Y_t, Z_t, K_t)$  satisfies (2). If it is necessary, we can choose a subsequence of (3). By Hölder's inequality,

$$E\left|\int_{t}^{T} \left(f\left(s, Y_{s}, Z_{s}\right) - f\left(s, Y_{s}^{n}, Z_{s}^{n}\right)\right) \mathrm{d}s\right|^{2}$$

$$\leq TE \int_{t}^{T} \left|\left(f\left(s, Y_{s}, Z_{s}\right) - f\left(s, Y_{s}^{n}, Z_{s}^{n}\right)\right)\right|^{2} \mathrm{d}s \longrightarrow 0.$$

$$(29)$$

From (27), we know

$$E\int_0^T |Y_t^n - Y_t|^2 dt \longrightarrow 0, \qquad (30)$$

and  $Y_t^n \to Y_t$ , a.e., so

$$\sqrt{E \int_{0}^{T} \left| Y_{t}^{n+1} - Y_{t}^{n} \right|^{2} \mathrm{d}t} \le \frac{1}{2^{n}}.$$
(31)

For any *n*,

$$\left|Y_{t}^{n}\right| \leq \left|Y_{t}^{1}\right| + \sum_{i=1}^{n-1} \left|Y_{t}^{i+1} - Y_{t}^{i}\right| \leq \left|Y_{t}^{1}\right| + \sum_{i=1}^{\infty} \left|Y_{t}^{i+1} - Y_{t}^{i}\right|.$$
 (32)

Then, we have

$$\sqrt{E} \int_{0}^{T} \sup_{n} |Y_{t}^{n}|^{2} dt 
\leq \sqrt{E} \int_{0}^{T} \left( |Y_{t}^{1}| + \sum_{i=1}^{\infty} |Y_{t}^{i+1} - Y_{t}^{i}| \right)^{2} dt 
\leq \sqrt{E} \int_{0}^{T} |Y_{t}^{1}|^{2} dt + \sum_{i=1}^{\infty} \sqrt{E} \int_{0}^{T} |Y_{t}^{i+1} - Y_{t}^{i}|^{2} dt 
\leq \sqrt{E} \int_{0}^{T} |Y_{t}^{1}|^{2} dt + \sum_{i=1}^{\infty} \frac{1}{2^{i}}.$$
(33)

From (H4), it follows

$$E \int_{0}^{T} \sup_{n} |f(s, Y_{s}, Z_{s}) - f(s, Y_{s}^{n}, Z_{s}^{n})|^{2} ds$$
  
$$\leq 2CE \int_{0}^{T} \left( \sup_{n} |Y_{s}^{n}|^{2} + |Y_{s}|^{2} + \sup_{n} |Z_{s}^{n}|^{2} + |Z_{s}|^{2} \right) ds < \infty.$$
(34)

Applying Lebesgue dominated convergence theorem, we deduce that  $(Y_t, Z_t, K_t)$  is the solution of (2) by continuity of the functions *f* and *g*.

Uniqueness. Let  $(Y_t^i, Z_t^i, K_t^i)$  (i = 1, 2) be two solutions of (2),  $\overline{Y}_t = Y_t^1 - Y_t^2$ ,  $\overline{Z}_t = Z_t^1 - Z_t^2$ . We apply Itô formula to  $e^{\beta t}\Psi_{M}(\overline{Y}_{t})$ , for any  $\beta \in \mathbb{R}$ ,

$$e^{\beta t} \Psi_{M}\left(\overline{Y}_{t}\right) + \beta \int_{t}^{T} e^{\beta s} \Psi_{M}\left(\overline{Y}_{s}\right) ds$$

$$+ \int_{t}^{T} e^{\beta s} \mathbf{1}_{\{-M \leq \overline{Y}_{s} \leq M\}} \left|\overline{Z}_{s}\right|^{2} ds$$

$$= \int_{t}^{T} e^{\beta s} \Psi_{M}'\left(\overline{Y}_{s}\right) \left(f\left(s, Y_{s}^{1}, Z_{s}^{1}\right) - f\left(s, Y_{s}^{2}, Z_{s}^{2}\right)\right) ds$$

$$+ \sum_{j=1}^{\infty} \int_{t}^{T} e^{\beta s} \mathbf{1}_{\{-M \leq \overline{Y}_{s} \leq M\}} \left|g_{j}\left(s, Y_{s}^{1}, Z_{s}^{1}\right) - g_{j}\left(s, Y_{s}^{2}, Z_{s}^{2}\right)\right|^{2} ds$$

$$- \sum_{j=1}^{\infty} \int_{t}^{T} e^{\beta s} \Psi_{M}'\left(\overline{Y_{s}}\right)$$

$$\times \left(g_{j}\left(s, Y_{s}^{1}, Z_{s}^{1}\right) - g_{j}\left(s, Y_{s}^{2}, Z_{s}^{2}\right)\right) d\beta_{j}(s)$$

$$- \int_{t}^{T} e^{\beta s} \Psi_{M}'\left(\overline{Y_{s}}\right) \overline{Z}_{s} dW_{s}$$

$$+ \int_{t}^{T} e^{\beta s} \Psi_{M}'\left(\overline{Y_{s}}\right) \left(dK_{s}^{1} - dK_{s}^{2}\right).$$
(35)

Taking expectation on both sides of (35),

$$\begin{split} Ee^{\beta t} \Psi_{M}\left(\overline{Y}_{t}\right) &+ \beta E \int_{t}^{T} e^{\beta s} \Psi_{M}\left(\overline{Y}_{s}\right) \mathrm{d}s \\ &+ E \int_{t}^{T} e^{\beta s} \mathbf{1}_{\{-M \leq \overline{Y}_{s} \leq M\}} \left|\overline{Z}_{s}\right|^{2} \mathrm{d}s \\ &\leq 2E \int_{t}^{T} e^{\beta s} \overline{Y}_{s} \left(f\left(s, Y_{s}^{1}, Z_{s}^{1}\right) - f\left(s, Y_{s}^{2}, Z_{s}^{2}\right)\right) \mathrm{d}s \\ &+ \sum_{j=1}^{\infty} E \int_{t}^{T} e^{\beta s} \mathbf{1}_{\{-M \leq \overline{Y}_{s} \leq M\}} \left|g_{j}\left(s, Y_{s}^{1}, Z_{s}^{1}\right) - g_{j}\left(s, Y_{s}^{2}, Z_{s}^{2}\right)\right|^{2} \mathrm{d}s \\ &\leq \left(\frac{2C}{1 - \sum_{j=1}^{\infty} \alpha_{j}} + \sum_{j=1}^{\infty} C_{j} + \frac{1 - \sum_{j=1}^{\infty} \alpha_{j}}{2}\right) E \int_{t}^{T} e^{\beta s} \left|\overline{Y_{s}}\right|^{2} \mathrm{d}s \\ &+ \frac{1 + \sum_{j=1}^{\infty} \alpha_{j}}{2} E \int_{t}^{T} e^{\beta s} \left|\overline{Z}_{t}\right|^{2} \mathrm{d}s. \end{split}$$
(36)

Let  $M \to \infty$ , and applying monotone convergence theorem, we have

$$Ee^{\beta t} \left| \overline{Y}_{t} \right|^{2} + \left( \beta - \frac{2C}{1-\alpha} - \sum_{j=1}^{\infty} C_{j} - \frac{1-\alpha}{2} \right) \\ \times E \int_{t}^{T} e^{\beta s} \left| \overline{Y}_{s} \right|^{2} ds \qquad (37) \\ + \frac{1-\alpha}{2} E \int_{t}^{T} e^{\beta s} \left| \overline{Z}_{s} \right|^{2} ds \leq 0.$$

When  $\beta$  is taken sufficiently large, we have  $\overline{Y}_t = 0$ , a.e., for all  $s \in [t, T]$ . So, we have  $\overline{Z}_t = 0$ , a.e. Then, we complete the proof. 

### Acknowledgments

The authors would like to take this chance to express their sincere gratitude to the National Natural Science Foundation of China (11201004 and 11371029), Natural Science Foundation of Anhui Province (KJ2011B176 and KJ2013B288), Professors(Doctors) Scientific Research Foundation of Suzhou University (2013jb04), and Foundation of Laboratory of Intelligent Information Processing of Suzhou University (2010YKF11).

### References

- [1] E. Pardoux and S. G. Peng, "Adapted solution of a backward stochastic differential equation," Systems & Control Letters, vol. 14, no. 1, pp. 55-61, 1990.
- [2] X. Mao, "Adapted solutions of backward stochastic differential equations with non-Lipschitz coefficients," Stochastic Processes and their Applications, vol. 58, no. 2, pp. 281-292, 1995.
- [3] K. Bahlali, E. H. Essaky, M. Hassani, and E. Pardoux, "Existence, uniqueness and stability of backward stochastic differential equations with locally monotone coefficient," Comptes Rendus Mathématique, vol. 335, no. 9, pp. 757-762, 2002.
- [4] P. Briand, B. Delyon, Y. Hu, E. Pardoux, and L. Stoica, " $L^p$  solutions of backward stochastic differential equations," Stochastic Processes and their Applications, vol. 108, no. 1, pp. 109–129, 2003.
- [5] J. P. Lepeltier and J. S. Martin, "Backward stochastic differential equations with continuous coefficient," Statistics & Probability Letters, vol. 14, pp. 55–514, 1997.
- [6] G. Jia, "A class of backward stochastic differential equations with discontinuous coefficients," Statistics & Probability Letters, vol. 78, no. 3, pp. 231-237, 2008.
- [7] N. El Karoui, S. Peng, and M. C. Quenez, "Backward stochastic differential equations in finance," Mathematical Finance, vol. 7, no. 1, pp. 1-71, 1997.
- [8] S. Hamadene and J.-P. Lepeltier, "Zero-sum stochastic differential games and backward equations," Systems & Control Letters, vol. 24, no. 4, pp. 259-263, 1995.
- [9] E. Pardoux and S. G. Peng, "Backward doubly stochastic differential equations and systems of quasilinear SPDEs," Probability Theory and Related Fields, vol. 98, no. 2, pp. 209-227, 1994.
- [10] B. Boufoussi, J. Casteren, and N. Mrhardy, "Generalized backward doubly stochastic differential equations and SPDEs with

5)

nonlinear Neumann boundary conditions," *Bernoulli*, vol. 13, no. 2, pp. 423-446, 2007.

- [11] N. El Karoui, C. Kapoudjian, E. Pardoux, S. Peng, and M. C. Quenez, "Reflected solutions of backward SDE's, and related obstacle problems for PDE's," *The Annals of Probability*, vol. 25, no. 2, pp. 702–737, 1997.
- [12] Y. Ren and L. Hu, "Reflected backward stochastic differential equations driven by Lévy processes," *Statistics & Probability Letters*, vol. 77, no. 15, pp. 1559–1566, 2007.
- [13] Y. Ren and M. El Otmani, "Generalized reflected BSDEs driven by a Lévy process and an obstacle problem for PDIEs with a nonlinear Neumann boundary condition," *Journal of Computational and Applied Mathematics*, vol. 233, no. 8, pp. 2027–2043, 2010.
- [14] Y. Ren, A. Lin, and L. Hu, "Stochastic PDIEs and backward doubly stochastic differential equations driven by Lévy processes," *Journal of Computational and Applied Mathematics*, vol. 223, no. 2, pp. 901–907, 2009.
- [15] S. Hamadène, "Reflected BSDE's with discontinuous barrier and application," *Stochastics and Stochastics Reports*, vol. 74, no. 3-4, pp. 571–596, 2002.

### **Research** Article

# An Improved Hybrid Genetic Algorithm with a New Local Search Procedure

### Wen Wan<sup>1</sup> and Jeffrey B. Birch<sup>2</sup>

<sup>1</sup> Department of Biostatistics, Virginia Commonwealth University, Richmond, VA 23298-0032, USA <sup>2</sup> Department of Statistics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0439, USA

Correspondence should be addressed to Wen Wan; wwan@vcu.edu

Received 9 January 2013; Accepted 26 August 2013

Academic Editor: Bin Wang

Copyright © 2013 W. Wan and J. B. Birch. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One important challenge of a hybrid genetic algorithm (HGA) (also called memetic algorithm) is the tradeoff between global and local searching (LS) as it is the case that the cost of an LS can be rather high. This paper proposes a novel, simplified, and efficient HGA with a new individual learning procedure that performs a LS only when the best offspring (solution) in the offspring population is also the best in the current parent population. Additionally, a new LS method is developed based on a three-directional search (TD), which is derivative-free and self-adaptive. The new HGA with two different LS methods (the TD and Neld-Mead simplex) is compared with a traditional HGA. Four benchmark functions are employed to illustrate the improvement of the proposed method with the new learning procedure. The results show that the new HGA greatly reduces the number of function evaluations and converges much faster to the global optimum than a traditional HGA. The TD local search method is a good choice in helping to locate a global "mountain" (or "valley") but may not perform the Nelder-Mead method in the final fine tuning toward the optimal solution.

### 1. Introduction

Genetic algorithms (GAs) perform well as a global search technique, but they may often take a relatively long time to converge to a global optimum [1–4]. Local search (LS) techniques have been incorporated into GAs to improve their performance through what could be termed as learning. Such HGAs, often known as memetic algorithms (MAs), were first introduced by Moscato [5, 6] and are viewed as a form of population-based genetic algorithms hybridized with an individual learning procedure capable of fine tuning the global search.

MAs represent one of the recent growing areas of research in evolutionary computation [7, 8]. Any population-based metaheuristic search method (inspired by Darwinian principles of natural selection) hybridized with any individual learning (inspired by Dawkins' notation "meme" [9]) procedure that belongs to the class of MAs [7]. In diverse contexts, MAs have also been referred to as hybrid evolutionary algorithms, Baldwinian evolutionary algorithms, Lamarkian evolutionary algorithms, cultural algorithms, or a genetic local search.

MAs have been successfully applied to hundreds of realworld problems in a wide range of domains [3, 7, 8, 10]. An important challenge of MAs is the tradeoff between global searching and local searching in terms of the time and computational effort [3, 10-14]; that is, the yet unanswered questions are when to apply a LS technique; to which individuals in the GA (or any other evolutionary algorithms) population should the LS technique be applied; and how much computational effort should be devoted to the LS technique. Recent literature presented several nonclassical MA methods that have been successful in reducing the total computational costs associated with an LS technique and that produce a profitable synergy from the hybridization of the GA (or any other evolutionary algorithms) and LS methods [7, 14–19]. But none of the nonclassical MAs are commonly accepted [10, 14]. Additionally, some of these methods, such as Seront and Bersini [15], Tang et al. [17], and Molina et al. [18, 19], may require the need for extra parameters.

Another challenge of MAs is the choice of successful LS techniques. Ning et al. [20] investigated the choice of LS techniques in HGAs and concluded that the choice affects

the search performance significantly and no single HGA always performs best on a diverse set of benchmark test functions.

In this study, to reduce the computational effort of an LS method without any extra parameters, a new HGA, called "a best-offspring HGA," denoted by BOHGA, is developed with a new individual learning procedure; that is, BOHGA performs an LS only when the best offspring (solution) in the offspring population is also the best in the current parent population. Additionally, a new LS method, a three-directional local search (TD), is introduced which is derivative-free and self-adaptive. The main idea of TD is that when the offspring performs better than both of its parents, three potential directions are constructed from parents to one of their offspring with a certain step length. We compare the new individual-learning HGA, BOHGA, with a traditional HGA, each using two memes: our TD method and the Neld-Mead simplex method. Both of these memes are derivative-free and suitable for real applications.

The remainder of this paper is organized as follows. We first briefly review the traditional GA and HGA. Our new HGA is introduced with its new individual learning procedure on when to perform the LS and on which offspring. We then present the two memes, respectively: one is the three-directional search (TD) and the other is the Nelder-Mead simplex meme. Through two benchmark functions, we present results for comparing the four HGAs for eight different settings of the GA operators and two different stopping rules. Finally, we present conclusions, discussions, and suggestions for future work.

### 2. The Genetic Algorithm and Hybrid Genetic Algorithm

Genetic algorithms (GAs) are iterative optimization procedures that repeatedly apply GA operators (such as selection, crossover, and mutation) to a group of solutions until some criterion of convergence has been satisfied. In a GA, a search point (solution), a setting in the search space with k dimensions (k variables), is coded into a string,  $\mathbf{x} = [x_1, \ldots, x_k]'$ , which is analogous to a *chromosome* in biological systems. The string/chromosome is composed of k characters,  $x_1, \ldots, x_k$ , which are analogous to the k genes. A set of multiple concurrent search points or a set of chromosomes (or individuals) is called a *population*. Each iterative step where a new population is obtained is called a generation. A GA hybridized with a local search procedure is called a hybrid genetic algorithm (HGA).

A basic HGA procedure has the following steps.

- Define an objective/fitness function, and set the GA operators (such as population size, parent/offspring ratio, selection method, number of crossovers, and mutation rate).
- (2) Randomly generate the initial population as the current parent population.
- (3) Evaluate the objective function for each individual (chromosome or solution) in the initial population.

- (4) Generate an offspring population by using GA operators (such as selection/mating, crossover, and mutation).
- (5) Evaluate the objective function of each individual in the offspring population.
- (6) Perform a local search on each offspring, evaluating fitness of each new location, and replace the offspring if there exists a locally improved solution.
- (7) Decide which individuals to include in the next population. This step is referred to as "replacement" in that individuals from the current parent population are "replaced" by a new population consisting of those individuals from the offspring and/or the parent populations.
- (8) If a stopping criterion is satisfied, then the procedure is halted. Otherwise, go to Step 4.

Without Step 6, an HGA is just a GA. Therefore, HGAs have all the properties possessed by GAs. Like GAs, HGAs are a large family of algorithms that have the same basic structure but differ from one another with respect to several strategies such as stopping rules, operators which control the search process, and the local search meme.

Based on previous experiences, in this study, we use a continuous HGA where chromosomes are coded as continuous measurement variables. Suppose there are k variables; that is, there are k genes in each chromosome. We also make the following assumptions. The (parent) population size is 2k and the offspring population size is also 2k. The type of selection we utilize is random pairing. The blending crossover is utilized and the number of crossover points depends on the number of dimensions of a specific objective function. Random uniform mutation is utilized and the mutation rate is set around or equal to 1/k. The type of replacement over both parent and offspring populations is either ranking or tournament. For details on the setting of the GA operators; see, for example, [21–25].

There are many choices of local search memes [20], two of which are used in this study. One meme is our newly developed "three-directional LS (TD)," introduced in Section 4. A second meme is a popular LS meme, the Nelder-Mead Simplex method, introduced in Section 5.

### 3. The Best-Offspring Hybrid Genetic Algorithm

As mentioned, our goal is to reduce the total costs associated with the LS. It has been noticed that the LS may be repeatedly performed on the same "mountain" (for finding a maximum) or "valley" (for finding a minimum) [15]. Therefore, it is possible that, after local searching, several chromosomes in a generation are very close to each other, standing on the same top of a mountain or at the same bottom of a valley. This may make it harder for the GA to maintain diversity in its population, an important consideration in avoiding converging to a local optimum [25]. Therefore, we propose the best-offspring HGA (BOHGA) where the LS is only performed on the best offspring in the offspring population when it is also the best overall chromosomes in the current parent population. When such a best offspring appears, it is very likely that the best offspring is located on a new, higher mountain or on a new lower valley. As will be soon demonstrated, this action tends to make BOHGA more computationally efficient and helps to prevent converging to a local optimum.

The general procedure for BOHGA is the same as that of HGA, except that in the *i*th generation we change Step 6 from the original HGA procedure into Steps 6.1–6.3 as follows.

- (6.1) Is the best offspring in the offspring population also the best over the current parent population?
- (6.2) If no, directly go to Step 7; that is, there is no LS in this generation.
- (6.3) If yes, then perform an LS on the best offspring considered as a starting point. Find the best locally improved solution and replace the best offspring by it. Then go to Step 7.

Actually, the BOHGA process is a special HGA process where an LS is not performed on every new offspring but only on the offspring which are best in both the offspring and the current parent populations. It is possible that not every generation of BOHGA requires an LS. The BOHGA procedure, therefore, strongly agrees with the original idea of MA, first introduced by Mascato in 1989 [5]; that is, initially let the GA explore a wide search space. Once a potential search solution is found by a GA, a fine tuning search will be conducted by an LS. Similar to both the GA and the HGA, the whole process is iterated until some appropriate stopping rule is satisfied.

### 4. A Three-Directional (TD) Meme

The idea of the TD meme is to construct three potential directions for an offspring whose performance is better than both of its parents in a generation. Thus, three paths are declared without requiring the gradient. When an offspring shows improvement from its parents in terms of the objective function, it may be possible to make continuous improvements by moving along the directions/paths from its parents to the offspring; that is, some search points are "collected" along the paths until no further improvement can be found. These parents can be considered as two different starting points. Both of their first steps from the two starting points go to the same point: the offspring. So two directions are established: one direction is from one of the parents to the offspring; the other is from the second of the parents to the offspring. Both directions have obtained improvement, since the best offspring of interest is an improvement over both its parents in terms of values of an objective function.

For example, consider a 2-dimensional (k = 2) problem along with the contours of a response (or values of an objective function) as illustrated in Figure 1. The offspring is denoted by O (expressed as  $\mathbf{x}_O = [x_{O1}, \ldots, x_{Ok}]'$ ) and its parents are denoted by P1 ( $\mathbf{x}_{P1} = [x_{P11}, \ldots, x_{P1k}]'$ ) and P2( $\mathbf{x}_{P2} = [x_{P21}, \ldots, x_{P2k}]'$ ). Obviously, there are two directions: one is from P1 to O, expressed as  $\delta_{P1O} = \mathbf{x}_O - \mathbf{x}_{P1} = [\delta_{11}, \delta_{12}, \ldots, \delta_{1k}]'$ , and the other is from P2 to O, expressed



3



FIGURE 1: A contour plot of a 2-dimensional problem with the three directions indicated: Parent 1 direction is from P1 to O; Parent 2 direction is from P2 to O; the common direction is a horizontal dotted line, starting at O towards the positive values on the  $X_1$  axis. The three "stars" represent the three points stopped on the three paths with no further improvement.

as  $\delta_{P2O} = \mathbf{x}_O - \mathbf{x}_{P2} = [\delta_{21}, \delta_{22}, \dots, \delta_{2k}]'$ . We refer to these two directions as Parent 1 and Parent 2 directions.

The third direction is the "common" direction, expressed as  $\delta = [\delta_{31}, \delta_{32}, \ldots, \delta_{3k}]'$ , and based on the two parent directions. If  $\delta_{1i}$  and  $\delta_{2i}$ , for  $i = 1, \ldots, k$ , are both positive (negative), then  $\delta_{3i}$  is positive (negative); that is, if both the parent directions are in common, say, both positive (negative) along the  $X_i$  axis, then the third direction is positive (negative) along the  $X_i$  axis. If  $\delta_{1i}$  and  $\delta_{2i}$ , for i = $1, \ldots, k$ , are opposite in direction, then  $\delta_{3i}$  is set to 0; that is, if the parent directions are not in common on the  $X_i$  axis. For more details on the three directions and determining their moving distances for each moving step, see the Appendix.

Figure 1 illustrates the three defined directions. The optimal point is denoted by " $\Theta$ ." It is easy to see the two parents directions, expressed as  $\delta_{P1O} = [\delta_{11}, \delta_{12}]'$  and  $\delta_{P2O} = [\delta_{21}, \delta_{22}]'$ , respectively. The third direction  $\delta = [\delta_{31}, \delta_{32}]'$ . Obviously,  $\delta_{31} > 0$  since both  $\delta_{11} > 0$  and  $\delta_{21} > 0$ ; that is, the common direction in this case is positive along the  $X_1$  axis. And  $\delta_{32} = 0$  since  $\delta_{12} > 0$  and  $\delta_{22} < 0$ ; that is, the common direction has no relative movement along the  $X_2$  axis.

Once the three directions are defined, starting at *O*, the TD method moves along the three directions/paths, with some appropriate step length for each moving step until no improvement is found in terms of an objective function. In Figure 1, the three "stars" on the paths denote that the three best points found on each path and the processes of moving along the paths will be stopped at their next points due to no further improvement.

The choice of the size of step length d depends on the degree of bumpiness of the surface of an objective function. We recommend that d should be in the physical range of 0.01 to 1.0. If the surface is very bumpy relative to the region of the domain, then the appropriate d should be relatively small. Otherwise, the appropriate d should be relatively large to make HGA more efficient.

In our BOHGA procedure, the TD meme will only be performed for the best offspring in the offspring population that is also the best in the current parent population. In our HGA procedure, the TD meme will be performed for those offspring whose performances are better than both of their parents. Since not every offspring performs better than either one of its parents, the TD meme will not be performed on every offspring, which is the major difference from a traditional HGA.

#### 5. Nelder-Mead Simplex Meme

The Nelder-Mead simplex method [26] is a very popular derivative-free method for finding a local minimum of a function [8]. For a two-dimensional problem, a simplex is a triangle, and the method is a pattern search that compares function values at the three vertices of a triangle. The worst vertex, where f(x, y) is largest, is rejected and replaced with a new vertex. A new triangle is formed and the search is continued. The process generates a sequence of triangles (which might have different shapes), for which the function values at the vertices get smaller and smaller. The size of the triangles is iteratively reduced and the coordinates of the minimum point are found. The simplex algorithm can easily be extended to higher dimensions [26]. In many numerical tests, the simplex method succeeds in obtaining a good reduction in the function value using a relatively small number of function evaluations but it is easy to converge to a local optimum and is generally not suitable for a highly nonlinear objective function [26].

Like the TD meme, the simplex meme requires a prespecified step length parameter, representing a guess of the problem's characteristic length scale. In this study, the step length parameter is set to the same size as d for the fair comparison between the simplex and TD memes. The C code for the simplex method is obtained from Numerical Recipes in C [27].

#### 6. Examples: Benchmark Functions

Using benchmark functions, our goal is to compare our BOHGA with a traditional HGA, with each procedure using one of the two LS techniques: our new TD method or the simplex method; that is, we compare the computational efficiency of four MAs: BOHGA with simplex (denoted as "BOHGA<sub>S</sub>"), BOHGA with TD ("BOHGA<sub>TD</sub>"), HGA with simplex ("HGA<sub>S</sub>"), and HGA with TD ("HGA<sub>TD</sub>") in computational efficiency for the four objective benchmark functions. As mentioned, HGA<sub>TD</sub> is different from the traditional HGA in that the TD local search will be performed only for those offspring whose performances are better than both their parents.

To make the comparisons comparable, the settings of the GA operators and the starting random numbers that are used to generate the initial populations are the same for each of the four MAs. In addition, since different starting random seeds may result in a different number of function evaluations to find an optimum, a Monte Carlo experiment is performed 100

times; that is, these four algorithms are run 100 times with 100 different starting random seeds. The four methods will be compared by averaging the results over the 100 replications of the experiment.

A different setting of GA operators may result in a different number of function evaluations. We choose 20 (k = 20) as a number of dimensions for the four benchmark functions. Therefore, as indicated in Section 2, both parent and offspring population sizes are 40. The number of crossover points is 4 or 8. The mutation rate is 0.05 (= 1/k) or 0.06, a slightly larger value than 1/k. The type of replacement over both parent and offspring populations is ranking or tournament. Therefore, there are a total of eight combinations of crossover, mutation, and replacement type; that is, there are eight GA settings used for comparisons.

Also two stopping rules are utilized for the experiment. The first stopping rule (rule 1) is that a method will be halted when a preset cut-off value (considered as a near-global optimum) is achieved. The cut-off value represents the user's best guess of the optimal value of the objective function. Rule 1 can be used to compare the computational efficiencies of the four methods in finding a near-global optimum of an objective function. The mean of a total number of function evaluations over 100 replications of each MA will be used for comparisons. Since sometimes the global and near-global optimal values are unknown, a second stopping rule (rule 2) is also considered. The second stopping rule is that a method will be halted at a preselected number of generations. Under rule 2, the number of function evaluations it takes for the four methods to converge to a global "mountain" or "valley" or even to a global optimum is compared; that is, the rate of convergence to a near-global or global optimum is compared across the four methods. Obviously, it is not relevant to compare the total number of function evaluations required given a fixed total number of generations. Graphs will be used to illustrate the comparisons of the four methods, by plotting mean best values of the objective function over 100 replications at each generation found by each method versus mean cumulative number of function evaluations at each generation by each algorithm. Four benchmark functions (the Rastrigin's, Schwefel's, Rosenbrock's, and Griewank's) are used for the comparisons, but, due to similar results and limited space, only the first two functions are presented as follows.

6.1. Comparisons for the Rastrigin's Function in 20 Dimensions. A generalized Rastrigin's function is given by

$$f(\mathbf{x}) = \sum_{i=1}^{k} \left( x_i^2 - 10 \cos(2\pi x_i) + 10 \right),$$
(1)
where  $-5.12 \le x_i \le 5.12$ ,

where *k* is the number of dimensions of the function (k = 20 in the study). Figure 2 shows its 1- and 2-dimensional surfaces. The surfaces are very bumpy in a narrow range (-5.12, 5.12). The goal is to find a minimal value and its corresponding location. The minimum of this function is known as min( $f(\mathbf{x})$ ) = f(0, ..., 0) = 0.0. From the left plot of



FIGURE 2: Surface of Rastrigin's function: (a) 1-dimension; (b) 2-dimension.

TABLE 1: Comparisons of BOHGA<sub>S</sub>, HGA<sub>S</sub>, BOHGA<sub>TD</sub>, and HGA<sub>TD</sub> in terms of mean number of evaluations under the eight settings of GA operators for the Rastrigin's function in 20 dimensions by stopping rule 1.

	8 settings of GA operators		Mean (evaluation)					
Replacement	Crossover	Mutation	BOHGA <sub>S</sub>	HGA <sub>S</sub>	BOHGA <sub>TD</sub>	$HGA_{TD}$		
	Λ	0.05	29174	510436	41352	39420		
Danking	4	0.06	29688	549620	46608	40174		
Ranking	0	0.05	33951	463661	30980	34720		
	0	0.06	30052	510260	35135	33627		
Tournament	4	0.05	62758	988761	132271	112930		
	4	0.06	113071	1424709	258692	265373		
	0	0.05	91747	1054834	214672	208569		
	0	0.06	212320	1538475	765658	880024		
Overall average			75345	880095	190671	201855		

Figure 2, a solution must be located on the global valley where the value of the objective function is less than about 1.0.

The step length for the TD meme is set to 0.05, the same value as for the simplex meme. The cut-off value used by rule 1, which is a near-global optimum, is set to 0.05. The preselected number of generations used by stopping rule 2 is 5,000.

Under stopping rule 1, Table 1 presents the mean number of function evaluations as a summary of the 100 repetitions for the Rastrigin's function in 20 dimensions for comparisons of the four algorithms. Table 1 shows that the number of evaluations required to obtain a value of the objective function is within 0.05 of the true minimum.  $BOHGA_s$ consistently performs the best with much smaller mean numbers of function evaluations than the  $BOHGA_{TD}$  and HGA<sub>TD</sub>, which are quite competitive to each other. In most of all the GA settings, BOHGA<sub>S</sub> has the smallest mean number of function evaluations, followed by BOHGA<sub>TD</sub>, HGA<sub>TD</sub>, and HGA<sub>s</sub>. In addition, the mean number of function evaluations greatly depends on the GA settings. The GA using ranking replacement obviously performs much better than the GA with tournament replacement in all of the four methods indicating that tournament replacement in MAs is not as efficient as ranking replacement. The mutation rate of 0.05 performs better than the rate of 0.06 in most cases.

Under stopping rule 2 with 5,000 generations, Figure 3 shows the mean best minimums of Rastrigin's function versus mean cumulative number of function evaluations at each generation over 100 replications by BOHGA<sub>s</sub>, HGA<sub>s</sub>, BOHGA<sub>TD</sub>, and HGA<sub>TD</sub>, respectively. The GA parameters were set at the ranking replacement, four crossover points, and 0.05 mutation rate. This figure illustrates that HGA<sub>s</sub> did not converge in the 50,000 mean function evaluations but the other three methods did converge. In the left plot of Figure 3, the BOHGA<sub>S</sub> procedure converged the fastest to the global "valley," followed closely by the BOHGA<sub>S</sub> and the HGA<sub>TD</sub> methods. The right plot in Figure 3, in an expanded scale, reveals that the BOHGA<sub>S</sub> procedure is actually the first to converge to the cutoff of 0.05 at about 38,000 mean cumulative function evaluations, followed by the  $HGA_{TD}$  and the BOHGA<sub>TD</sub> methods. It is clear that these three methods have very similar behavior for this function.

6.2. Comparisons for the Schwefel's Function in 20 Dimensions. A generalized Schwefel function from Schwefel [28] is given by

$$\sum_{i=1}^{k} -x_{i} \sin\left(\sqrt{|x_{i}|}\right), \text{ where } -500 \le x_{i} \le 500, \quad (2)$$



FIGURE 3: Best minimums of Rastrigin's function in 20 dimensions versus number of function evaluations at each generation averaged over 100 replications of the four MA methods under the GA setting (ranking replacement, 4 crossover points, and 0.05 mutation rate). (a) Overall view in a full scale; (b) highlighted view in an expanded scale.



FIGURE 4: Surface of Schwefel's function: (a) 1-dimension; (b) 2-dimension.

where *k* is the number of dimensions of the function. The minimum of the objective function is given by  $\min(f(\mathbf{x})) = f(420.9687, \ldots, 420.9687)$ . The minimum is dependent on *k*, the number of dimensions. When k = 20, the minimum value is -8,379.66. Figure 4 shows the 1- and 2-dimensional surfaces for the Schwefel function. In the left plot of the figure, a solution must be located in the deepest valley, when value of the objective function is less than about -300.0 in the 1-dimensional case.

Although the Schwefel function has a nonlinear bumpy surface, its surface is relatively smooth in a range (-500, 500) when compared to the surface of the Rastrigin's function. The step length for the TD meme is set to 0.5, the same as for the

simplex meme. The preselected number of generations used by stopping rule 1 is 1,000. The cut-off near-global value is set to -8,379.0.

Similar to Table 1, under stopping rule 1, Table 2 presents the mean total number of function evaluations as a summary of the 100 repetitions for the Schwefel's function for comparison of the four algorithms. Table 2 shows that the numbers of evaluations required to obtain a value of the objective function smaller than -8,379.0 by BOHGA<sub>S</sub>, BOHGA<sub>TD</sub>, and HGA<sub>TD</sub> are all consistently much less than required by HGA<sub>S</sub> overall settings. BOHGA<sub>S</sub> consistently performs the best with much smaller mean numbers of function evaluations than the HGA<sub>TD</sub> and BOHGA<sub>TD</sub>, which

TABLE 2: Comparisons of  $BOHGA_S$ ,  $HGA_S$ ,  $BOHGA_{TD}$ , and  $HGA_{TD}$  in terms of mean of the number of evaluations under the eight settings of GA operators for the Schwefel's function in 20 dimensions by stopping rule 1.

8 settings of GA operators			Mean (evaluation)					
Crossover	Mutation	BOHGA <sub>S</sub>	HGA <sub>S</sub>	BOHGA <sub>TD</sub>	$HGA_{TD}$			
4	0.05	13595	471668	26792	31243			
Ţ	0.06	15049	471251	26972	31590			
8	0.05	13230	518101	20588	28070			
0	0.06	13792	546366	20207	29972			
4	0.05	28631	1059412	47893	132294			
Ţ	0.06	37991	1404750	94763	281186			
Q	0.05	37792	1408634	74805	221730			
0	0.06	57270	1824465	214563	815384			
Overall average		27169	963081	65823	196434			
	8 settings of GA operators Crossover 4 8 4 8 4 8 0verall a	8 settings of GA operators Crossover Mutation 4 0.05 0.06 0.06 0.06 4 0.05 0.06 0.06 0.06 0.06 0.06 0.06	8 settings of GA operators Crossover Mutation BOHGA <sub>s</sub> 4 0.05 13595 0.06 15049 13230 13250 135	8 settings of GA operators         Mean (ergenter)           Crossover         Mutation         BOHGA <sub>S</sub> HGA <sub>S</sub> 4         0.05         13595         471668           4         0.06         15049         471251           8         0.05         13230         518101           8         0.06         13792         546366           4         0.05         28631         1059412           4         0.06         37991         1404750           8         0.05         37792         1408634           0.06         57270         1824465	8 settings of GA operators       Mean (evaluation)         Crossover       Mutation       BOHGA <sub>s</sub> HGA <sub>s</sub> BOHGA <sub>TD</sub> A       0.05       13595       471668       26792         A       0.06       15049       471251       26972         B       0.05       13230       518101       20588         B       0.06       13792       546366       20207         A       0.05       28631       1059412       47893         A       0.06       37991       1404750       94763         B       0.05       37792       1408634       74805         B       0.06       57270       1824465       214563			



FIGURE 5: Best minimums of Schwefel's function in 20 dimensions versus number of function evaluations at each generation averaged over 100 replications of the four MA methods under the GA setting (ranking replacement, four crossover points, and 0.05 mutation rate). (a) Overall view in a full scale; (b) highlighted view in an expanded scale.

are quite competitive to each other. Overall the GA settings, BOHGA<sub>S</sub> has the smallest mean numbers of function evaluations, followed by BOHGA<sub>TD</sub>, then HGA<sub>TD</sub>, and finally HGA<sub>S</sub>. In addition, the GA setting with ranking replacement performs much better than with tournament replacement in all of the four methods. This again indicates that tournament replacement in MAs is not as efficient as ranking replacement. The mutation rate of 0.05 performs better than the rate of 0.06 in most cases.

Similar to Figure 3, under stopping rule 2 with 1,000 generations, Figure 5 shows the mean best minimum of the Schwefel's function versus mean cumulative number of function evaluations at each generation over 100 replications obtained by each MA in the GA setting with the ranking replacement, four crossover points, and 0.05 mutation rate.

The left plot of Figure 5 shows that HGA<sub>S</sub> is the slowest to converge while BOHGA<sub>TD</sub> and BOHGA<sub>S</sub> have converged to a global "valley" at a similar yet faster rate than HGA<sub>TD</sub>. The right plot of Figure 5, in an expanded scale, shows in detail that BOHGA<sub>S</sub> is the fastest to converge to the cutoff of -8379.0 at about 1,400 mean cumulative function evaluations, followed by the BOHGA<sub>TD</sub>, followed by the HGA<sub>TD</sub>.

### 7. Conclusion and Discussion

The importance of memetic algorithms in both real-world applications and academic research has lead to the establishment of the series of international Workshops On Memetic Algorithms (WOMA) and a dedicated book [13]. From these workshops, the following important questions are raised: (1) when to apply local improvement heuristics, (2) to which individuals in the evolutionary algorithms population should local searches be applied, and (3) how much computational efforts to devote to local search algorithms. These questions remain unanswered, and more research effort is required to gain the understanding and insights that may lead to guidelines for the design of efficient and effective algorithms [13].

This paper presents an improved and simplified MA, BOHGA, with a novel individual learning procedure on when to perform a local search (or individual learning). Unlike a classical MA/HGA procedure, where a local search is performed on each offspring (solution), our new MA performs a local search when the best offspring is also found to be the best among the current parent population. This new learning procedure does not require any extra parameters.

We also develop a new meme, a three-directional local search, TD, which is derivative-free and self-adaptive. The main idea of TD is that three potential directions are constructed from parents to their offspring with a certain step length, when the offspring performs better than both of its parents.

The four well-known benchmark functions with very different experimental ranges are used to compare BOHGA<sub>s</sub>, HGA<sub>S</sub>, BOHGA<sub>TD</sub>, and HGA<sub>TD</sub>. The results under stopping rule 1 (that an algorithm is halted when a near-globaloptimum cutoff is achieved) match the results under stopping rule 2 (that an algorithm is halted at a preselected number of generations) for both functions. These results indicate that BOHGA with the new individual learning procedure works much more efficiently than the traditional HGA, whichever meme is chosen.  $\mathrm{HGA}_\mathrm{TD}$  , where the LS is performed only on those offspring that have an objective function values superior to both of their parents, is quite competitive to BOHGA. These results also indicate that the TD meme is likely to help algorithms converge faster to a global "valley" but does not appear to converge as quickly during the final fine tuning stage as the simplex meme. The results from the Rosenbrock's and the Griewank's functions (which are not presented here) are similar to those from the Rastrigin's and the Schwefel's functions.

During the comparisons of the four MAs, we used eight different settings of GA operators and found that ranking replacement performs uniformly better than tournament replacement for both functions. The mutation rate of 0.05 (which is 1/k, k = 20 in both of the benchmark functions) performed better than the rate of 0.06 in most cases. The different number of crossover points had no obvious effect on the number of function evaluations.

In summary, our new HGA with an individual learning procedure performs a LS only when the best offspring is also the best within the parent population. Our new HGA not only reduces the number of function evaluations required by the LS, but also improves accuracy and efficiency in finding an optimal solution. The TD meme is a good choice in helping finding a global "valley" or "peak" but may not perform as well as the Nelder-Mead method at the final fine tuning. It is noted that our HGA has combined our new meme with a GA. We speculate that our new procedure would also be effective when combined with other evolutionary algorithms.

Several issues remain for further study. For example, the three derivative-free directions defined in the TD meme may not be optimal. Another issue concerns the appropriate step length, once the directions are chosen. The size of a step length, arbitrarily chosen by us, may affect the efficiency of the MAs. We found that the TD may converge faster to a global "valley" or "peak" than the simplex meme but may be not as fast at finding an optimum at the fine tuning stage. In a future study, we may combine the TD and simplex memes together, using TD first to reach the global "valley" or "peak," followed by the simplex meme to fine tune the solution. A further issue involves the optimal settings of the GA operators. In this study, the three main GA operators: the type of replacement, the number of crossover points, and the mutation rate, have been studied. However, there may be some other operators affecting the GA performance, such as the population size and the parent/offspring ratio. We plan to study these issues in future work.

C++ code is available upon request from the authors.

### Appendix

### Mathematical Representation of the Three-Direction: A Local Search

We first introduce our notation. Parent 1 (*P*1) is given by  $\mathbf{x}_{P1} = [x_{P11}, \ldots, x_{P1k}]'$ , where **x** is a vector of size  $k \times 1$  where k is the number of factors or the number of dimensions. Similarly, Parent 2 (*P*2) is given by  $\mathbf{x}_{P2} = [x_{P21}, \ldots, x_{P2k}]'$ , and their offspring (*O*) is expressed as  $\mathbf{x}_O = [x_{O1}, \ldots, x_{Ok}]'$ . Parent 1 direction (from *P*1 to *O*) is expressed as  $\delta_{P1O}$  and Parent 2 direction (from *P*2 to *O*) is as  $\delta_{P2O}$ . And the common direction is simply denoted as  $\delta$ . The new points after the first step along the three directions are expressed as  $\mathbf{x}_{New1} = [x_{New11}, \ldots, x_{New1k}]'$ ,  $\mathbf{x}_{New2} = [x_{New21}, \ldots, x_{New2k}]'$ , and  $\mathbf{x}_{New} = [x_{New1}, \ldots, x_{Newk}]'$ , corresponding to Parent 1, Parent 2, and their common direction, respectively. The appropriate moving distance on each axis in each moving step is expressed as d.

Parent 1 direction, which essentially is the different distances on each dimension between points *P*1 and *O*, is expressed as

$$\delta_{P1O} = \mathbf{x}_O - \mathbf{x}_{P1} = [\delta_{11}, \delta_{12}, \dots, \delta_{1k}]'.$$
(A.1)

Similarly, the Parent 2 direction is expressed as

$$\delta_{P2O} = \mathbf{x}_O - \mathbf{x}_{P2} = [\delta_{21}, \delta_{22}, \dots, \delta_{2k}]'.$$
(A.2)

To keep the same directions and move along the three paths, the moving distance on each axis should be in constant proportion to each other, as the method of steepest ascent/descent in response surface methodology (RSM). (In RSM, the constant proportion on the *i*th dimension is defined as  $\hat{\beta}_i/\hat{\beta}^{\times}$ , where the  $\hat{\beta}_i$  is the *i*th estimated coefficient in the estimated first-order model and the  $\hat{\beta}^*$  is the largest coefficient in magnitude among the *k* estimated coefficients,
that is,  $\hat{\beta}^* = \max_{i=1,...,k} |\hat{\beta}_i|$ .) From this ratio, we can see that the proportion only depends on the  $\beta_i$ , the *i*th coefficient. The moving distance on the *i*th dimension is defined as  $(\hat{\beta}_i/\hat{\beta}^*) * \rho$ , where the  $\rho$  is an appropriate fixed distance. (For more details, please see Myers and Montgomery [29, Pages 205–207]).

In our GA application, the main idea in moving along the Parent 1 path is the same as that in the method of steepest ascent/descent; that is, to keep the constant proportion in each dimension and move some appropriate fixed distance (which is *d* in our case) along Parent 1 path. But the difference between our GA case and RSM is the starting point. In the GA case, the starting points are *P*1 and *P*2, not *O*; that is, the first step has already been completed. So the next moving step starts at *O*. The largest moving distance in the first step is also not *d*, but  $\max_{i=1,...,k} |\delta_{1i}|$ , where the  $\delta_{1i}$  is the moving distance on *i*th axis in (A.1). Let  $\delta_1^*$  denote  $\max_{i=1,...,k} |\delta_{1i}|$ . In our study, if  $\delta_1^* < d$ , then the moving distant in the next step will be  $\delta_1^*$ . Otherwise, the distance in the next moving distance.

The procedure of moving along the Parent 1 direction is as follows.

- Calculate δ<sub>P1O</sub> and then find δ<sub>1</sub><sup>\*</sup> = max<sub>i=1,...,k</sub>|δ<sub>1i</sub>|, the largest distance in the first moving step.
- (2) If  $\delta_1^* < d$ , then the next new position on the *i*th axis, i = 1, ..., k, is defined as  $x_{\text{Newl}i} = x_{Oi} + (\delta_{1i}/\delta_1^*) \times d$ . Otherwise, the new position is  $x_{\text{Newl}i} = x_{Oi} + \delta_{1i}$ .
- (3) Check the region of the new point  $\mathbf{x}_{New1} = [x_{New11}, \dots, x_{New1k}]'$ . If  $x_{New1i}$  is greater than its upper bound (which is the largest value in the *i*th domain), then let it be the upper bound. Similarly, if it is less than its lower bound (which is the lower value in the *i*th domain), then let it be the lower bound. (Usually, the upper bounds and lower bounds have been given through defining the objective function.)
- (4) Evaluate the new point  $\mathbf{x}_{\text{New1}}$  by the objective function. If the new point performs worse than the point  $\mathbf{x}_{\text{O}}$ , then the process of moving along the Parent 1 direction is halted. If the new point performs better than the  $\mathbf{x}_{\text{O}}$ , then replace the point  $\mathbf{x}_{\text{New1}}$  by the next new point  $\mathbf{x}_{\text{New1}} + \mathbf{\Delta}_{N1O}$ , where  $\mathbf{\Delta}_{N1O} = \mathbf{x}_{\text{New1}} \mathbf{x}_{\text{O}}$ . (The "N1O" means "New point from Parent 1" to "Offspring.") Then return to Step 3.

The procedure for moving along the Parent 2 direction is the same as that for the Parent 1 direction. However, the procedure for the common direction is slightly different from them, due to the different starting points. The starting points from the parents directions are P1 or P2, while the starting point in the common direction is O.

As mentioned earlier, building the common direction depends on whether both parent directions are consistent or not. If they are consistent on *i*th axis (either both positive or both negative), then move the same direction on the *i*th axis as the parent directions. Otherwise, stay on that axis without any movement, due to inconsistent directions. There is a special case: one of the moving distances on an axis in the parent directions is zero and the other is nonzero. In this case, we recommend movement in the same direction with the parent direction with nonzero moving distance on the axis.

The procedure for movement along the common direction is as follows.

- (1) Calculate  $\delta_{P1O}$  and  $\delta_{P2O}$  as (A.1) and (A.2).
- (2) The next new point is defined as  $\mathbf{x}_{New} = [x_{New1}, \dots, x_{Newk}]'$  along the path from the common direction. To establish the common direction, three situations on each axis/dimension are possible: (a) the  $\delta_{1i} \times \delta_{2i} > 0$  which means that there is a common direction on the *i*th axis; (b) The  $\delta_{1i} \times \delta_{2i} < 0$  which means that there is not a common direction on the *i*th axis; and (c) the  $\delta_{1i} \times \delta_{2i} = 0$  which means that at least one of  $\delta_{1i}$  and  $\delta_{2i}$  equals zero.
  - (2.1) If the situation is (a), then the new point position on the *i*th axis is given by  $x_{\text{New}i} = x_{Oi} + \min(|\delta_{1i}|, |\delta_{2i}|, d)$  if both  $\delta_{1i}$  and  $\delta_{2i}$  are positive, or  $x_{\text{New}i} = x_{Oi} - \min(|\delta_{1i}|, |\delta_{2i}|, d)$  if both  $\delta_{1i}$  and  $\delta_{2i}$  are negative.
  - (2.2) If the situation is (b), the new point position on the *i*th axis is given by  $x_{\text{New}i} = x_{Oi}$  (no movement on the *i*th axis in this situation).
  - (2.3) If the situation is (c), there are three subcases: (1)  $\delta_{1i} = 0$  and  $\delta_{2i} \neq 0$ ; (2)  $\delta_{1i} \neq 0$  and  $\delta_{2i} = 0$ ; and (3)  $\delta_{1i} = 0$  and  $\delta_{2i} = 0$ .
    - (2.3.1) For case (1), if  $|\delta_{2i}| \ge d$ , then  $x_{\text{New}i} = x_{\text{O}i} + d$ (when  $\delta_{2i} > 0$ ) or  $x_{\text{New}i} = x_{\text{O}i} - d$  (when  $\delta_{2i} < 0$ ). Otherwise,  $x_{\text{New}i} = x_{\text{O}i} + \delta_{2i}$ .
    - (2.3.2) For case (2), similar to case (1), if  $|\delta_{1i}| \ge d$ , then  $x_{\text{New}i} = x_{Oi} \pm d$ . Otherwise  $x_{\text{New}i} = x_{Oi} + \delta_{1i}$ .
    - (2.3.3) For case (3),  $x_{\text{New}i} = x_{Oi}$ .
- (3) Check the range of the new point  $\mathbf{x}_{\text{New}}$ .
- (4) Evaluate the point  $\mathbf{x}_{\text{New}}$ . If the new point performs worse than the point  $\mathbf{x}_{\text{O}}$ , then the process for moving along the common direction is stopped. If the new point is better than  $\mathbf{x}_{\text{O}}$ , then replace the point  $\mathbf{x}_{\text{New}}$ by the next new point  $\mathbf{x}_{\text{New}} + \boldsymbol{\Delta}_{\text{NCO}}$ , where  $\boldsymbol{\Delta}_{\text{NCO}} = \mathbf{x}_{\text{New}} - \mathbf{x}_{\text{O}}$ . (The "NCO" means "New from Common directions" and "Offspring"). Return to Step 3.

### Acknowledgments

The authors wish to thank the anonymous referees and the editors for their constructive comments on an earlier draft of this paper.

#### References

- L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, NY, USA, 1991.
- [2] D. E. Goldberg and S. Voessner, "Optimizing global-local search hybrids," in *Proceedings of the 1st International Conference of*

*Genetic and Evolutionary Computation (GECCO '99)*, pp. 220–228, 1999.

- [3] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, "Realcoded memetic algorithms with crossover hill-climbing," *Evolutionary Computation*, vol. 12, no. 3, pp. 273–302, 2004.
- [4] Z. Michalwicz, Genetic Algorithms + Data Structure = Evolution Programs, AI, Springer, New York, NY, USA, 3rd edition, 1996.
- [5] P. A. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms," Tech. Rep. 826, Caltech Concurrent Computation Program, 1989.
- [6] P. A. Moscato, "Memetic algorithms: a short introduction," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glower, Eds., p. 219234, McGraw-Hill, London, UK, 1999.
- [7] Y.-S. Ong, N. Krasnogor, and H. Ishibuchi, "Special issue on memetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 1, pp. 2–5, 2007.
- [8] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multifacet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.
- [9] R. Dawkins, *The Selfish Gene*, Oxford University Press, New York, NY, USA, 1990.
- [10] D. Sudholt, "Local search in evolutionary algorithms: the impact of the local search frequency," in *Algorithms and computation*, vol. 4288 of *Lecture Notes in Computer Science*, pp. 359– 368, Springer, Berlin, Germany, 2006.
- [11] N. Krasnogor and J. E. Smith, "Emergence of profitable search strategies based on a simple inheritance mechanism," in *Proceedings of the International Conference on Genetic and Evolutionary Computation*, pp. 432–439, Morgan Kaufmann, San Mateo, Calif, USA, 2001.
- [12] M. Lozano, F. Herrera, and J. R. Cano, "Replacement strategies to maintain useful diversity insteady-state genetic algorithms," *Information Sciences*, vol. 178, pp. 4421–4433, 2008.
- [13] W. E. Hart, N. Krasnogor, and J. E. Smith, "Editorial introduction special issue on memetic algorithms," *Evolutionary Computation*, vol. 12, no. 3, 2004.
- [14] T. A. El-Mihoub, A. A. Hopgood, L. Nolle, and A. Battersby, "Hybrid genetic algorithms: a review," *Engineering Letters*, vol. 13, pp. 2–11, 2006.
- [15] G. Seront and H. Bersini, "A new GA local search hybrid for continuous optimization based on multi-level single linkage clustering," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '00)*, pp. 90–95, Morgan Kaufmann, Las Vegas, Nev, USA, 2000.
- [16] M. Lozano and C. García-Martínez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report," *Computers & Operations Research*, vol. 37, no. 3, pp. 481–497, 2010.
- [17] K. Tang, Y. Mei, and X. Yao, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1151–1166, 2009.
- [18] D. Molina, M. Lozano, and F. Herrera, "Memetic algorithm with local search chaining for continuous optimization problems: a scalability test," in *Proceedings of the 9th International Conference on Intelligent Systems Design and Applications (ISDA '09)*, pp. 1068–1073, IEEE Computer Society, December 2009.
- [19] D. Molina, M. Lozano, and F. Herrera, "MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization," in *Proceedings of the 6th IEEE*

World Congress on Computational Intelligence (WCCI '10), July 2010.

- [20] Z. Ning, Y. S. Ong, K. W. Wong, and M. H. Lim, "Choice of memes in memetic algorithm," in *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS '03)*, 2003.
- [21] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Mass, USA, 1989.
- [22] M. Hamada, H. F. Martz, C. S. Reese, and A. G. Wilson, "Finding near-optimal Bayesian experimental designs via genetic algorithms," *The American Statistician*, vol. 55, no. 3, pp. 175–181, 2001.
- [23] D. G. Mayer, J. A. Belward, and K. Burrage, "Robust parameter settings of evolutionary algorithms for the optimisation of agricultural systems models," *Agricultural Systems*, vol. 69, no. 3, pp. 199–213, 2001.
- [24] F. Ortiz Jr., J. R. Simpson, J. J. Pignatiello Jr., and A. Heredia-Langner, "A genetic algorithm approach to multiple-response optimization," *Journal of Quality Technology*, vol. 36, no. 4, pp. 432–450, 2004.
- [25] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, Second edition, 2004, With 1 CD-ROM (Windows).
- [26] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.
- [27] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.
- [28] H. P. Schwefel, Evolution and Optimum Seeking, John Wiley & Sons, New York, NY, USA, 1995.
- [29] R. H. Myers and D. C. Montgomery, Response Surface Methodology: Process and Product Optimization Using Designed Experiments, John Wiley & Sons, 2002.

### Research Article

## Layer-Based Data Aggregation and Performance Analysis in Wireless Sensor Networks

### Hongju Cheng,<sup>1</sup> Yongzhao Chen,<sup>1</sup> Naixue Xiong,<sup>2</sup> and Feifei Li<sup>1</sup>

<sup>1</sup> College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China <sup>2</sup> School of Computer Science, Colorado Technical University, Colorado, CO 80907, USA

Correspondence should be addressed to Naixue Xiong; nxiong@coloradotech.edu

Received 3 June 2013; Accepted 24 July 2013

Academic Editor: Bin Wang

Copyright © 2013 Hongju Cheng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the similarity and correlation among sensed data in wireless sensor network, it is an important way to reduce the number of packets transmitted with data aggregation technology so as to prolong the network lifetime. However, data aggregation is still a challenging issue since quality-of-service, such as end-to-end delay, is generally considered as a severe criterion required in many applications. We focus on the minimum-latency data aggregation problem and proposed a new efficient scheme for it. The basic idea is that we first build an aggregation tree by ordering nodes into layers, and then we proposed a scheduling algorithm on the basis of the aggregation tree to determine the transmission time slots for all nodes in the network with collision avoiding. We have proved that the upper bound for data aggregation with our proposed scheme is bounded by ( $15R + \Delta - 15$ ) for wireless sensor networks in two-dimensional space. Extensive simulation results have demonstrated that the proposed scheme has better practical performance compared with related works.

### 1. Introduction

Recent advances in microelectronics, low-power embedded modulators, and wireless networking have led to the emergence of the wireless sensor network [1]. All these sensor nodes are self-organized and cooperated in similar way as the ad hoc network. Such characteristics make it possible to deploy sensor network to obtain information about the covered area in an inaccessible location. It is expected that sensor networks open new vistas for many potential applications. Data aggregation [2] is generally considered as an important method in the sensor networks by aggregating and forwarding the raw data which is originated from multiple sources. In this way, the data collection from all nodes in the network is in fact done with the aggregation tree, in which the sink serves as the root and the nonleaf nodes will aggregate all raw data from their children and forward the result to their parents. This helps to minimize the traffic in the network, reduce the energy consumption, and increase the network lifetime accordingly.

Quality-of-Service (QoS), such as end-to-end delay, is generally considered as a severe criterion required in many

applications. When data is collected by local nodes in the network, the sensed data is generally required to reach the root within a given time delay. To provide end-toend delay guarantee is pioneering and challenging with the data aggregation problem in wireless sensor networks due to two separate observations. Firstly, the nonleaf nodes on the aggregation tree cannot forward to their parents until they have gathered all data from their descendants, which is possible to increase end-to-end delay especially in case that the aggregation tree is rather high. Secondly, due to the broadcasting characteristic of wireless communication, the exposed/hidden problem is the main factor which not only reduces capacity of the wireless network but also increases one-hop transmission delay. A careful scheduling algorithm concerned with the transmission time slots for all nodes in the network is rather necessary and important for data aggregation problem aiming at providing minimum end-toend delay while utilizing the data aggregation scheme and eliminating the above collision problem.

The minimum-latency data aggregation problem in two-dimensional wireless sensor networks is well studied and proved to be NP hard [3]. Chen et al. had designed a  $\Delta$ —approximation algorithm for this problem, where  $\Delta$  denotes the maximum node degree in the network [3]. Huang et al. introduced a constant approximation algorithm with the latency bounded by  $(23R + \Delta - 18)$  [4] by building the aggregation tree with the help of maximum independent set (MIS), where *R* denotes the network radius. Wang et al. improved the idea of Huang and proposed a scheduling algorithm with the latency bound as  $(15R + \Delta - 15)$  [5]. In this work, we have proved that the theoretic bound for data aggregation with our proposed scheme is also  $(15R + \Delta - 15)$  for wireless sensor networks in two-dimensional space However, the proposed scheme has better performance compared with the previously mentioned related works [4, 5] especially in dense networks.

With the rapid development of applications in threedimensional networks, such as underwater acoustic monitoring [6, 7], underground tunnels [8], space sensor network [9], and aerosphere pollution monitoring [10], it is interesting and challenging to study the minimum-latency data aggregation problem with three-dimensional wireless sensor networks too. Most of the current works in three-dimensional sensor networks aimed at providing connectivity, coverage, or location solutions [11]. In this work, we have also extended our efficient scheme to the case in three-dimensional network. We also demonstrated its practical performance compared with works originated from the two-dimensional networks.

The rest of this paper is organized as follows. we introduce the related works in Section 2 and present preliminary in Section 3. Section 4 introduces a new minimum-latency data aggregation (MDA) scheme. Section 5 has analyzed the correctness and performance of MDA scheme. In Sections 6 and 7, we present simulation results and conclusion.

### 2. Related Works

Data aggregation is considered as an important solution for the wireless sensor networks. The main goal of dataaggregation algorithms is to gather and aggregate data in an energy-efficient manner so that network lifetime is enhanced [2]. Krishnamachari et al. illustrated the impact of data aggregation by comparing its performance with traditional end-toend routing schemes [12]. The optimal data aggregation problem was proven to be NP hard, and heuristic algorithms were proposed to gather data from multiple sources to the sink.

The process of aggregating data could reduce the transmission of data in the network, hence to reduce energy consumption. Most previous works have mainly focused on energy-saving issue and it has been investigated in [12, 13]. Wireless sensor networks often use tree topologies. This is not only because tree's structure is suitable for a network with one sink node, but also their simplicity is very attractive when network resources are limited. There are some papers that build a data aggregation tree to control the delay [3–5, 14–22].

Broadcast and data aggregation are the most fundamental and useful applications in wireless sensor networks. Data aggregation is sometimes referred to converge cast. Annamalai et al. designed a heuristic algorithm [14] which constructs a tree for both broadcast and converge cast. Simulation results show that latency of this algorithm is very high, and this algorithm has high requirements for hardware of wireless sensor. Upadhyayula et al. [15] designed a heuristic algorithm for converge cast alone and purposed to reduce the latency and energy consumption. This algorithm constructs a tree using a greedy approach where new nodes are added to the tree so that weight on the branch to which it is added is less.

Chen et al. first proved that the minimum data aggregation time problem is NP hard and designed a  $\Delta$  approximation algorithm for this problem [3], where  $\Delta$ denotes the maximum node degree in the network. This algorithm is centralized and has a high latency, which makes this algorithm impractical. In paper [16], the authors focused on the latency performance of data aggregation and considered applications for which the total delay of the sensed data is important instead of the maximum delay. Wan et al. [17] designed a distributed algorithm to construct a dominating tree. The algorithm is to construct a maximal independent set and then select connected nodes to construct a dominating tree. Wan et al. [18] constructed three aggregation schedules of latency  $(15R + \Delta - 4)$ ,  $(2R + O(\log R) + \Delta)$ , and  $((1 + <math>O(\log R/\sqrt[3]{R}))R + \Delta)$ .

Huang et al. proposed a scheduling algorithm with the latency bound  $(23R + \Delta - 18)$  [4], where R is the network radius and  $\Delta$  is the maximum node degree. However, this algorithm has obvious errors in the first-fit algorithm so that the schedules are not collision-free in many cases. We will modify this algorithm and compare with it in simulation. Yu et al. proposed a distributed scheduling algorithm, named DAS, the latency bound  $(24D + 6\Delta + 16)$  [19], where D is the network diameter. Xu et al. [20, 21] constructed a data aggregation tree using an existing approach with a small modification. Then, they presented an efficiently centralized algorithm and a distributed scheduling implementation. They proved that the latency bound is at most  $(16R + \Delta - 14)$ . They focused on data aggregation scheduling problem and proved that the lower latency bound under any interference model is max{log n, R} in paper [20], where n is the network size. Wang et al. designed a scheduling algorithm, Peony-treebased data aggregation (PDA), and proved the latency bound is  $(15R + \Delta - 15)$  [5].

Some works [22–27] concerned with the delay control. Yuan et al. designed a Multi-level Fusion Synchronization Protocol to achieve the desired trade-offs between the credibility and the aggregation latency [22]. Some works have investigated the energy latency tradeoff [23–25]. Given a deadline, they minimized the amount of missed data [23], minimized overall energy dissipation of sensor nodes [24], or minimized the maximum energy consumption [25]. Many applications of sensor networks require real-time data aggregation. Chipara et al. proposed dynamic conflict-free query scheduling [26], a novel scheduling technology based on TDMA, which is a natural choice for real-time sensor network applications.

Event-triggered data aggregation refers to no node need to send data until a relevant, unpredicted event occurs in the network. A distributed TDMA scheduling protocol for data aggregation is proposed in paper [27], which is called DATP. The sensor nodes send dummy packets in order to determine whether they are interfered with each other, if not, they are assigned the same time slot.

In this paper, we aimed at the minimum latency data aggregation problem with the data aggregation tree. We designed a novel approach to build the aggregation tree, which selects the maximum independent set in even layers. We have proved that the upper bound for data aggregation with our proposed scheme is bounded by  $(15R + \Delta - 15)$  for wireless sensor networks in two-dimensional space. We also simulated the case in three-dimensional wireless sensor networks. The results have demonstrated that the proposed scheme has better performance compared with related works.

### 3. Preliminary

In this section, we will describe the system model, related terms and give a detailed problem definition.

3.1. Network Model. We consider a wireless sensor network with a sink node *s* and *N* sensor nodes. Each node is equipped with a wireless radio by which a node can receive/transmit data packet. Assume that all nodes have the same transmission range  $r_c$ ; the topology of wireless sensor network can be represented as an undirected graph G = (V, E), where *V* is the set of all nodes and  $E \subseteq V^2$  is the set of undirected links. *E* is defined as  $E = \{(u, v) \in V^2 \mid d_{uv} \leq r_c\}$ , in which  $d_{uv}$  is the Euclidean distance between node *u* and *v*,  $r_c$  is the transmission radius.

3.2. Interference Model. Here, we use the symbol time slot to denote the period that one node is used to send or receive data packet. The radio generally works in halfduplex mode and thus one node cannot transmit and receive packet simultaneously due to broadcast characteristic of wireless communication, which is generally named as the hidden/exposed terminal problems. Here, we assume that the interference range is identical to the transmission range [28].

#### 3.3. Related Terms

*3.3.1. Data Aggregation.* We assume that the network is designed with simple data aggregation function, such as max, min, and average. In this way, the final result after aggregation generally has the same property as the incoming originally data, such as packet length and prior. And all these results can be sent or received during one single time slot operation.

3.3.2. Independent Set (IS). Given an undirected graph G = (V, E) and  $I \subseteq V$ , I is called an independent set if any two nodes in I are not adjacent. It is obvious that two nodes  $u, v \in$  IS denotes that both u and v are not within the transmission range of each other. A maximum independent set is used to represent the independent set with maximum number of nodes for a given graph G.

3.3.3. Connected Dominating Set (CDS). Given an undirected graph G = (V, E) and  $D \subseteq V$ , D is called a dominating set if

D induces a connected subgraph of G and every node in G either belongs to D or is adjacent to a node in D.

*3.3.4. Concurrent Set (CS).* The concurrent set denotes a set of nodes, in which each node can transmit without conflict with the transmission of other nodes.

3.4. Data Aggregation Scheduling. With the terms mentioned previously, the data aggregation scheduling problem in the wireless sensor networks is defined to find a sequence of concurrent set  $S_1, S_2, \ldots, S_T$  so that the latency is minimized, where  $\bigcup_{i=1}^T S_i = V - \{s\}$ . For any  $i \neq j, S_i \cap S_j = \emptyset$ . This problem has been proved to be NP hard [14].

### 4. Proposed Scheme

In this section, we give the detailed description of our proposed scheme for the minimum-latency data aggregation (MDA) problem. The basic idea is that we firstly construct a data aggregation tree by dividing the nodes into layers and then design a scheduling scheme in which each node is assigned with one time slot for transmission while collision is avoided. The notations used in the algorithm are summarized in Notations.

#### 4.1. Data Aggregation Tree Construction

4.1.1. Initialization. For a given network graph, initially we can divide all nodes into R layers with the breadth-first searching algorithm which starts from the sink. The sink node s is in the layer 0.  $L_i$  denotes the set of the nodes that are *i*-hop away from the sink s. Here we use  $L_i$  to denote the set of nodes which are *i*-hop away from the sink s. The sink node s is in the layer 0.

4.1.2. Independent Set Construction for Each Layer. Here, we use the independent set to construct the data aggregation tree. The idea is described as follows. Initially, we add the sink node *s* to the independent set in layer 0. Obviously, there is only node in layer 0, and no conflict occurs. Let  $D_0 = \{s\}$ , in which  $D_0$  denotes the independent set of layer 0. Generally, we select independent set  $D_i$  in layer *i* if *i* is an even, and the process starts from layer 0. In layer *i*, we check each node in sequence to find whether it conflicts with nodes in set  $D_j$  (j = 0, 2, ..., i - 2). If not, the node is added to  $D_i$ ; otherwise, we will move it to layer (i + 1). The corresponding pseudocode is given in Algorithm 1. After the independent sets  $D_i$  (i = 0, 2, 4, ...) have been selected, we can obtain the total independent set as  $D = \cup D_i$  (i = 0, 2, 4, ...).

4.1.3. Construct the Aggregation Tree. The basic idea is that we construct a connected dominating set as the aggregation tree based on the previous independent set. Obviously, sink node s in layer 0 is the root of data aggregation tree. We add s to  $D_0$ . We find that  $D_2$  with Algorithm 1 and the nonindependent nodes are moved to layer 3. Each node u in layer i, i > 2 can check its neighbors in upper layer and adds them to

**Input:** Graph *G*, *L*<sub>1</sub>, *L*<sub>2</sub>, ..., *L*<sub>*R*</sub>, layer number *i*; **Output:** Maximum Independent set *D<sub>i</sub>*, *D*, and *L<sub>i</sub>*. (1) *D<sub>i</sub>* = Ø; (2) for each node  $u \in L_i$ ; (3) if  $u \notin \{x \mid (x, v) \in E, v \in D\}$ , then (4) *D<sub>i</sub>* = *D<sub>i</sub>*  $\cup \{u\}$ ; (5) else *L<sub>i</sub>* = *L<sub>i</sub>* - {*u*}; *L<sub>i+1</sub>* = *L<sub>i+1</sub>*  $\cup \{u\}$ ; (6) end if (7) end for

ALGORITHM 1: Layer independent set construction.

the set upper(u). Node u is moved to layer (i + 1) in case that |upper(u)| = 0. In each even layer, we build the layer maximum independent set and move other nodes down to the next layer. The next step is to build the children list for each node. Firstly, node u in layer (i-1) checks the neighbors in layer i and adds the neighbor to the set lower(u). Repeat the previous process and the data aggregation tree can be built finally.

From the process of aggregation tree construction, we can clearly see that the children of nodes in IS are either leaf nodes or connecting nodes, while the children of the connecting node are nodes in IS. The pseudocode of the data aggregation tree construction process is presented in Algorithm 2.

4.2. Data Aggregation Scheduling. Nodes are scheduled according to their roles on the tree. We firstly schedule leaf nodes and then the nodes in connected dominating set.

4.2.1. Aggregation of Leaf Nodes. In this process, we are to assign time slots for each leaf nodes on the data aggregation tree, and the leaf nodes can send data packet to their parent during the assigned time slots. In order to avoid interference and minimize the latency at this phase, leaf nodes are divided into k concurrent sets and nodes in each set can transmit data simultaneously with interference avoided, which is denoted as  $S_1, S_2, \ldots, S_k$ , where k is the number of the concurrent sets. Nodes in S<sub>i</sub> are scheduled to transmit data in the *i*th time slot. In the following, we will introduce that the process leaf nodes are separated into different concurrent sets. Firstly, there is no concurrent sets and k = 0. Then, we choose a node x from  $L_R$  and try to insert x inserted into one concurrent set. Obviously, we need to create a new set and let k = 1, and we can insert x into  $S_1$ . Then, we pick another node y in layer R and determine whether it conflicts with any node in current concurrent set  $S_i$  (i = 1, 2, ..., k) or not. If  $S_i$  is found, y is inserted into it; otherwise, a new concurrent set is necessary, and we have k increased by one and insert y into the new set. This process continues until all leaf nodes are assigned to a set.

The pseudocode of concurrent set construction is presented in Algorithm 3.

4.2.2. Aggregation of Connected Dominating Nodes. When all leaf nodes are assigned time slots to transmit data packet to their parents, we are to schedule the time slots for

nonleaf nodes on the aggregation tree. Starting from the bottom of the tree, the algorithm assigns the sending time to all connected dominating nodes. According to the tree construction process, the dominating nodes are located in even layers and connecting nodes in odd layer. Hence, the algorithm will construct concurrent set in every layer. That is, the dominating nodes in layer *d* are divided into  $m_d$  sets if *d* is even. Similarly, the connected nodes in odd layer *c* are also divided into  $n_c$  sets. Due to the fact that the latency of the aggregation of leaf nodes is *k*, we schedule all connected dominating nodes from the tree bottom to the root. Generally, for each node  $u \in L_i$ , if *u* is assigned to the *j*th set in layer *i*, the scheduled time slot for *u* is  $(k + j + \sum_{d=H}^{i-2} m_d + \sum_{c=H}^{i-1} n_c)$  in case that *i* is even, where *H* denotes the depth of the data aggregation tree. And the result is  $(k + j + \sum_{c=H}^{i-2} n_c + \sum_{d=H}^{i-1} m_d)$  in case that *i* is odd.

4.2.3. Data Aggregation Scheduling Algorithm. After the previous two steps, the sending time of all nodes in the network are assigned. The pseudocode of the data aggregation scheduling is presented in Algorithm 4. As we can observe from the algorithm details, the time of the sink *s* received is latency =  $k + \sum_{i=H}^{i=1} (m_i + n_i) + 1$ . Particularly, we have  $n_i = 0$  in case that mod(i, 2) = 1, and otherwise  $m_i = 0$ .

4.3. Example Demonstration. In this section, we use an example to demonstrate the process of data aggregation tree construction in detail. Figure 1(a) shows the topology of a random network with 20 nodes, in which each node is represented by a cycle, and the node identification is marked below the cycle. There is a link/edge between two nodes if they are within the transmission range of each other. Here, we assume that the sink node; that is, *s* is located at the center of the network.

Initially, we can organize the network topology into layers, Algorithm 1, and nodes in each layer can be observed in Figure 1(b). For example, there are only one node, that is, *s*, in layer 0, and five nodes, that is, 1, 2, 3, 4, and 5 in layer 2. There are totally 5 layers in this example with the network radius as 4. It is obvious that the layer number for each node denotes the distance from the node to the sink.

The second process is to build the MIS for each even layer. The process is carried out in increasing order of the layers. Firstly, there is only one node in layer 0, and the node *s* is added to the independent set for layer 0, that is,  $D_0 = \{s\}$ , which is marked as black in Figure 1(b). Note that nodes in layer 1 are one hop away from node *s*, and thus they are not possible to be included into the IS, and they are added into the aggregation tree with their parent as *s*. The next step is to select IS for layer 2. With Algorithm 2, we can obtain  $D_2 = \{6, 8, 9, 11, 12, 13\}$  which is also marked black in the figure; 7, 10, and 14 are dependent nodes and they are moved down to layer 3 (the process is simulated via dash line in Figure 1; then we can select parent nodes form nodes in  $D_2$ , which are marked as gray in Figure 1(b). In this way, the tree construction process for previous three layers is finished.

Now, consider nodes in layer 3. Note that 7, 10, and 14 are moved from previous layer to layer 3, and it is possible

**Input:** Graph G = (V, E); **Output**: Data aggregation tree T = (V', E'), the dominated set  $D_i$ , i = 2, 4, ..., the set of connective nodes  $C_i$ , i = 1, 3, ...(1) V' = V,  $E' = \emptyset$ ; (2) Breadth first search graph *G* with the root as *s*; (3) Divide all node of V' into layers  $L_0, L_1, \ldots, L_R$ ; (4)  $D_1 = \{s\};$ (5) for each node  $x \in L_1$ , par(x) = s;  $E' = E' \cup \{(x, s)\}$ ; (6) for i = 2 to R do (7)calculate  $upper(u), u \in L_i$ if |upper(u)| = 0 and  $u \in L_i$ , then  $L_i = L_i/u$ ,  $L_{i+1} = L_{i+1} \cup \{u\}$ ; (8)if *i* is even, construct the layer independent set  $D_i$ , and sequence the *lower*(*y*) with decreasing order of the set size as (9)  $y_1, y_2, \dots, y \in L_{i-1}$ , and the set size as  $y_1, y_2, \dots, y \in L_{i-1}$ ; (10)j = 1;(11)while  $L_i$  != empty do for each node  $u \in lower(y_i)$ (12) $par(u) = y_j, E' = E' \cup \{(u, y_j)\}, L_i = L_i/u;$ (13)(14)if *i* is even then  $C_i = C_i \cup y_i$ ; (15)end for j = j + 1;(16)end while (17)(18) end for



```
Input: Data aggregation tree T, G;
Output: Concurrent Set S_1, S_2, \ldots, S_k.
(1) initialize S_i = \emptyset, i = 0, 1, ...;
(2) for each layer i from R to 1;
(3) select one leaf node u from layer i;
(4)
      j = 1;
(5) if S_i = \emptyset, then S_i = S_k \cup \{u\}, go to step 3;
(6) for each v \in S_i,
          if (u, par(v)) \notin E and par(u) \neq par(v), then S_i = S_i \cup \{u\}, break;
(7)
(8)
       end for
(9)
      if u \notin S_i, then j = j+1, go to step 5;
(10) end for
```

ALGORITHM 3: Concurrent set construction.

```
Input: Data aggregation tree T = (V', E'), depth of the data aggregation tree H;
Output: Sending time t(u) for each node u, and latency.
(1) for each node u \notin CDS, run Algorithm 3 to construct the concurrent set S_1, S_2, \ldots, S_k;
(2) if u \in S_i then t(u) = j;
(3) latency = k;
(4) for i = H to 1
(5)
     if i is even, then
(6)
         all nodes u \in L_i, construct the concurrent set SD_1, SD_2, \ldots, SD_m;
(7)
         if u \in SD_j, then t(u) = latency + j, latency = latency + m;
(8)
      else
(9)
          all nodes u \in L_i, construct the concurrent set SC_1, SC_2, \ldots, SC_n;
(10)
          if v \in SC_i, then t(v) = latency + j, latency = latency + n;
(11)
      end if
(12) end for
```



(d) Data aggregation tree

FIGURE 1: An example to demonstrate the process of MDA algorithm.

that some nodes in layer 3 cannot find their parents in upper layer, that is, layer 2. With Algorithm 2, we calculate the set upper for each node in layer 3, and we have |upper(17)| = 0, |upper(22)| = 0, |upper(23)| = 0. As we can observe from Figure 1(b), 17 can connect to sink s via 7; however, 17 cannot find proper parent in layer 2 since 7 is moved to layer 3. The case is similar to 22 and 23, and all these three nodes are moved down to layer 4, which is marked with dash line in Figure 1(b). After these processes are done, we can select parent for nodes in layer 3 and add them into the tree.

There are five nodes, that is, 17, 22, 23, 24, and 25 in layer 4. We firstly check whether they can find their parents in upper layers before building the independent set. It can be seen that set upper is not empty for all nodes in layer 4; secondarily, we build the MIS for layer 4, and  $D_4 = \{17, 22, 24, 25\}$  which is marked black; 23 is moved down to layer 5; finally, we choose

parent for nodes in layer 4, and 7, 14, 19, and 20 are selected accordingly and included in set  $C_3$  which is marked gray. In this way, the tree construction for former four layers is finished which is illustrated in Figure 1(c).

Note that there is only 23 in layer 5 with parent 22 in layer 4, and it can be inserted into the tree directly. So far, all nodes in the network are included in the aggregation tree with the height as 5. The final result is demonstrated in Figure 1(d).

The last process is to schedule the aggregation process on the tree with Algorithms 3 and 4. Following the idea of our scheduling scheme, the leaf nodes on the tree are scheduled firstly. It starts from the final layer, and all leaf nodes are scheduled into collision-free time slots. As we can see from Figures 1(a) and 1(d), these leaf nodes can be divided into two separate sets, {23, 15, 10, 21, 5} and {16, 18}. Nodes in the first set are assigned to the first time slot, and nodes in the second set are assigned to second time slot. In this way, we use only 2 time slots to ensure that all leaf nodes can send to their parents while collision is avoided. Then, we are to schedule the nonleaf nodes on the tree. It can be seen that nodes in  $D_4 = \{17, 24, 22, 25\}$  can be scheduled with one time slot, that is, slot 3. The process is carried out in the same way until all nodes are scheduled. And we can see that in time slot 11 the sink node can collect all data from all nodes in the network.

### 5. Performance Analysis

**Lemma 1.** Given the original network radius R, the depth of the data aggregation tree by our MDA algorithm does not exceed (2R - 1).

*Proof.* As shown in the process of the tree construction, nodes in layer 0 and layer 1 are not moved down.

In layer 2, we select the layer maximum independent set  $D_2$  and move the nonindependent nodes to layer 3.  $L'_2$ denotes the set of the nodes which moved from layer 2 to layer 3. Since each node in  $L'_2$  is adjacent to at least one node in  $D_2$ , it finds its parent in  $D_2$ . Layer 3 is the base for nodes in original layer 2.

In layer 3, some nodes may not find their neighbors in  $D_2$ . Their neighbors have been moved to the set  $L'_2$  in the same layer with them. Nodes in layer 3 cannot find their neighbors as their parent in upper layer, which should be moved to layer 4.  $L'_3$  denotes the set of nodes which move from layer 3 to layer 4. Hence, each node in  $L'_3$  can find its neighbor as its parent in  $L'_2$ . We should select the layer maximum independent set  $D_4$ in layer 4, so some nodes in  $L'_3$  may be as the nonindependent nodes moved to layer 5.  $L''_3$  denotes the set of nodes which move from layer 3 to layer 5. Since each node in  $L'_3$  is adjacent to one node in  $D_4$  at least, it finds its parent in  $D_4$ . Layer 5 is the lowest layer the nodes in original layer 3 can be moved.

In layer 4, because there may be some nodes whose neighbors are all in the set  $L''_3$ , they should be moved to layer 6 with the nodes in  $L''_3$  moving to layer 5.  $L'_4$  denotes the set of nodes which moves from layer 4 to layer 6. We should select the layer maximum independent set  $D_6$  in layer 6, so some nodes in  $L'_4$  may be as the nonindependent nodes moved to layer 7.  $L''_4$  denotes the set of nodes which moves from layer 4 to layer 7. Because each node in  $L''_4$  is adjacent to at least one node in  $D_6$ , it finds its parent in  $D_6$ . Layer 7 is the base layer that nodes in original layer 4 can be moved.

After analyzing the nodes moving cases in the four layers, we can get the number of layers each node is displaced from its original position depending on its neighbor in upper layer. When a node was moved to an even layer, the layer independent set should be selected in this even layer, so the node may be as a nonindependent node moved to the next layer. Then, the node in the odd layer can find its neighbor in the maximum independent set; the node will no longer be moved. In this way, it would move into its final position in an odd layer. Therefore, in the worst case, a node's moving layer is the moving layer of its neighbor in upper layer plus 2. For a node *u* in layer *k*, if all its neighbors in layer (k - 1) have been

moved to layer (2(k-1)-1), at worst, they have to be moved to layer (2k-3). We can infer that the final layer which *u* moved to is equal to the sum of the layer (2k - 3) which neighbors of *u* moved to plus 2, that is, layer 2(k - 1). Given the original network radius *R*, in the worst case, the nodes in layer *R* may be moved to layer (2R - 1).

**Lemma 2.** The latency of aggregation from leaf nodes is  $\Delta - 1$ .

*Proof.* Given the maximum node degree  $\Delta$ , the latency of aggregation from leaf nodes is  $(\Delta - 1)$  [4, 5].

**Theorem 3.** *The latency bound of the connected dominating nodes aggregation is* 15R - 14*.* 

*Proof.* Now, we will estimate the data aggregation of connected dominating nodes. According to Lemma 1, the depth of data aggregation tree is at most (2R - 1). In the tree, the number of layers of the dominative nodes except the sink *s* is (R-1). The number of layers of the connective nodes is (R-1), because connective nodes cannot be in layer *R*. According to [4, 5], the latency bound of data aggregation from dominative nodes to connective nodes is 4, and the latency bound of data aggregation from connective nodes is 11. Particulary, the sink node *s* is the root of the tree. Thus, connective nodes in layer 1 are its children and it takes at most 12 time slots to finish the transmission.

Based on the previous analysis, the latency bound of the connected dominating nodes aggregation is 4(R-1) + 11(R-2) + 12 = 15R - 14.

**Theorem 4.** The total latency bound of data aggregation is  $15R + \Delta - 15$ .

*Proof.* According to the MDA, the total latency is the sum of latencies of leaf nodes aggregation and connected dominating nodes aggregation. The total latency bound of data aggregation is  $(\Delta - 1) + (15R - 14) = 15R + \Delta - 15$ .

**Theorem 5.** The time complexity with the MDA algorithm is  $O(Rn^2 + n\Delta)$ , in which n is the node number, R is the network radius, and  $\Delta$  is the maximum node degree.

*Proof.* Initially, we use the breadth search algorithm to construct the layer structure for a given network, and the complexity with Algorithm 1 is  $O(n^2)$ .

During the tree construction process of Algorithm 2, we check each node whether they can find their parent, and the time complexity is  $O(n^2)$ . Secondarily, we select independent set for even layers with the rest nodes excluded in the IS moved down, and the time complexity for this operation is O(n). Finally, each node will select its parent and be added into the aggregation tree; the time complexity is  $O(n^2)$ . Assume the network radius as *R*, the time complexity for the tree construction is  $O(Rn^2)$ .

During the scheduling process, we first schedule the leaf nodes and then nonleaf nodes. Note that we schedule these nodes from layers far away from the sink, and the collision conflict occurs only in case that the transmission is carried



FIGURE 2: Simulation results in two-dimensional networks with different network sizes.

out simultaneously by neighbors of its parent, and thus the scheduling time complexity is  $O(n\Delta)$ .

In this way, the time complexity with our MDA algorithm is  $O(n^2) + O(n^2) + O(Rn^2) + O(n\Delta) = O(Rn^2)$ .

Based on the previous analysis, we could know that the total latency bound of data aggregation obtained by MDA is the same as [5] in two-dimensional space, which is the best result we have ever known. Besides, our proposed algorithm could achieve better performance by reducing the height of the aggregation tree without increasing the time complexity. The later experimental results also validate the efficiency of the method.

### 6. Simulation Results

Our simulation is accomplished by generating a random wireless sensor network in MATLAB software. We evaluate the performance of the proposed MDA and related in twoand three-dimensional networks.

6.1. Simulation Results in Two-Dimensional Networks. In this simulation part, the network topology is randomly generated

by placing nodes in a fixed region of size  $100 \text{ m} \times 100 \text{ m}$ . We compare our MDA with the algorithm proposed by Huang et al. in [15] (denoted as HUANG in short in the figures) and PDA proposed by Wang in [19].

6.1.1. Impact of Network Size. The first group of simulations estimates the impact of network size. The transmission range of each sensor is fixed to 10 m. The aggregation latency is measured when the network size varies from 300 to 800. We compare its average performance by building 11 different network topologies.

Figure 2(a) compares the network radius after constructing the data aggregation tree by using these algorithms. As mentioned in the previous section, the worst case for the tree height with the MDA algorithm is 2R - 1. However, it can be seen that the upper bound is seldom met in the simulations. This is because nodes are not always moved during the tree construction process with our MDA algorithm, which leads to the reduction of tree height. However, the results with HUANG and PDA are almost always 2R - 1. It is seen that the network radius of HUANG and PDA is approximately twice as much as MDA. Our algorithm shows great improvement on the tree height as we can observe from Figure 2(a).



FIGURE 3: Simulation results in two-dimensional networks with different communication ranges.

Figure 2(b) shows the simulation result of aggregation latency of leaf nodes. Due to the fact that independent set is selected only in even layers with our MDA, the number of connected dominating nodes is less than the other two algorithms, and thus the number of leaf nodes is larger than the other two algorithms. So, it is reasonable that, in some case, the delay for leaf nodes scheduling costs more time slots with our MDA algorithm, which can be seen in case that network size is 850.

Figure 2(c) compares the aggregation latency of nodes in the connected dominating sets. As mentioned previously, the MDA algorithm only selects dominating nodes in even layers, and thus it leads to less number of dominating nodes compared with the other two algorithms. And accordingly, the required number of time slots for aggregation is generally smaller.

The total latency for the data aggregation process is calculated with the previous two parts. As we can see from Figure 2(d), our MDA algorithm runs better than HUANG and PDA, which is more significant in case that the network size is very large. 6.1.2. Impact of Communication Range. The second group of simulations estimates the impact of communication range. The network size is fixed to 1000 while the communication range varies from 6 m to 15 m. Figure 3(a) shows the network radius after constructing the data aggregation tree by using HUANG, PDA, and MDA. With the communication range becoming larger, the node degree increases, and accordingly the radius decreases. Figure 3(c) compares the latency of aggregation from leaf nodes. The latency of aggregation from leaf nodes decreases when the communication range becomes larger. The total latency is the sum of latencies of leaf nodes aggregation and connected dominating nodes aggregation. The smaller the transmission range, the more independent nodes, the more connected dominating nodes accordingly. Hence, the total latency is mainly determined by the latency of aggregation from connected dominating nodes. When the transmission range becomes larger, the node degree increases; thus, the number of leaf nodes increases. Now, the total latency depends on latency of aggregation from leaf nodes. The latency is large at both ends of curves and relative small in middle, as shown in Figure 3(d).



FIGURE 4: Simulation results in three-dimensional networks with different network sizes.

6.2. Simulation Results in Three-Dimensional Networks. In the part, the network topology is randomly generated by placing nodes in a  $100 \times 100 \times 100 \text{ m}^3$  cube. We compare our MDA with HUANG and PDA when applied to the three-dimensional networks.

6.2.1. Impact of Network Size. The third group of simulations estimates the impact of network size in three-dimensional networks. Similar to the first group of simulations, the results of this group are shown in Figure 4.

6.2.2. Impact of Communication Range. The last group of simulations estimates the impact of communication range in three-dimensional networks. Similar to the first group of simulations, the results of this group are shown in Figure 5.

### 7. Conclusions

Data aggregation is an import technology used to reduce the energy consumption in the wireless sensor networks. In this paper, we focused on the minimum latency data aggregation problem in wireless sensor networks and proposed a novel minimum-latency data aggregation (MDA) algorithm to build the aggregation tree as well as scheduling scheme for the node transmission in the network. We proved that the theoretical latency bound for MDA in the plane is  $(15R + \Delta -$ 15). We have also simulated the case in three-dimensional wireless sensor networks. Extensive simulation results have demonstrated that our algorithm has good performance compared with the related algorithms. In the future work, we are to develop a distributed version of the proposed MDA algorithm with energy considered by constructing a load balance aggregation tree. Furthermore, we will extend our work to the multisink wireless sensor networks.

### Notations

dist(u, v): Euclidean distance between u and v

- N(u): The set of neighbors for node u
- *R*: The network radius
- *H*: Depth of the data aggregation tree
- $\Delta$ : Maximum node degree
- $L_i$ : The set of nodes that are *i*-hop away from the sink *s*
- upper(u): The set of neighbors in upper layer of node u



FIGURE 5: Simulation results in two-dimensional networks with different communication ranges.

- *lower*(u): The set of neighbors in lower layer of node u
- *tag*(*u*): The tag whether *u*'s sending time is determined
- *par(u)*: The parent node of node *u* on the data aggregation tree.

### Acknowledgments

This work is supported by the National Science Foundation of China under Grants nos. 61370210, 61103175, the Fujian Provincial Natural Science Foundation of China under Grants nos. 2011J01345, 2013J01232, and the Development Foundation of Educational Committee of Fujian Province under Grant no. 2012JA12027.

### References

- I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [2] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.

- [3] X. Chen, X. Hu, J. Zhu, X. Jia, J. Wu, and Y. He, "Minimum data aggregation time problem in wireless sensor networks," in *Mobile Ad-Hoc and Sensor Networks*, vol. 3794, pp. 133–142, Springer, Berlin, Germany, 2005.
- [4] S. C.-H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. Yao, "Nearly constant approximation for data aggregation scheduling in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM* '07), pp. 366–372, May 2007.
- [5] P. Wang, Y. He, and L. Huang, "Near optimal scheduling of data aggregation in wireless sensor networks," *Ad Hoc Networks*, vol. 11, pp. 1287–1296, 2013.
- [6] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.
- [7] D. Pompili and T. Melodia, "Three-dimensional routing in underwater acoustic sensor networks," in *Proceedings of the* 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05), pp. 214–221, ACM, Montreal, Canada, October 2005.
- [8] D. Wu, L. Bao, and R. Li, "A holistic approach to wireless sensor network routing in underground tunnel environments," *Computer Communications*, vol. 33, no. 13, pp. 1566–1573, 2010.

- [9] X. Hong, M. Gerla, R. Bagrodia, T. J. Kwon, P. Estabrook, and G. Pei, "The mars sensor network: efficient, energy aware communications," in *Proceedings of the IEEE Military Communications Conference (MILCOM '01)*, vol. 1, pp. 418–422, October 2001.
- [10] W. Tsujita, A. Yoshino, H. Ishida, and T. Moriizumi, "Gas sensor network for air-pollution monitoring," *Sensors and Actuators B*, vol. 110, no. 2, pp. 304–311, 2005.
- [11] V. Ravelomanana, "Extremal properties of three-dimensional sensor networks with applications," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 246–257, 2004.
- [12] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings* of the 22nd International Conference on Distributed Computing Systems Workshops, pp. 575–578, 2002.
- [13] Y. Wu, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: NPcompleteness and approximation algorithm," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1013–1021, April 2008.
- [14] V. Annamalai, S. K. S. Gupta, and L. Schwiebert, "On tree-based convergecasting in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference* (WCNC '03), vol. 3, pp. 1942–1947, 2003.
- [15] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta, "A lowlatency and energy-efficient algorithm for convergecast in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 6, pp. 3525–3530, December 2003.
- [16] C. Joo, J.-G. Choi, and N. B. Shroff, "Delay performance of scheduling with data aggregation in wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, March 2010.
- [17] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 3, pp. 1597–1604, June 2002.
- [18] P.-J. Wan, S. C.-H. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks," in *Proceedings of the 10th ACM International Symposium* on Mobile Ad Hoc Networking and Computing (MobiHoc '09), pp. 185–193, ACM, New Orleans, La, USA, May 2009.
- [19] B. Yu, J. Li, and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 2159–2167, April 2009.
- [20] X. Xu, X. Y. Li, X. Mao, S. Tang, and S. Wang, "A delayefficient algorithm for data aggregation in multihop wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 163–175, 2011.
- [21] X. Xu, S. Wang, X. Mao, S. Tang, and X. Li, "An improved approximation algorithm for data aggregation in multi-hop wireless sensor networks," in *Proceedings of the 2nd ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC '09)*, pp. 47–56, New Orleans, La, USA, May 2009.
- [22] W. Yuan, S. V. Krishnamurthy, and S. K. Tripathi, "Synchronization of multiple levels of data fusion in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 1, pp. 221–225, December 2003.

- [23] S. Hariharan and N. B. Shroff, "Maximizing aggregated revenue in sensor networks under deadline constraints," in *Proceedings* of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (CDC/CCC '09), pp. 4846–4851, December 2009.
- [24] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 244–255, March 2004.
- [25] L. Becchetti, P. Korteweg, A. Marchetti-Spaccamela, M. Skutella, L. Stougie, and A. Vitaletti, "Latency constrained aggregation in sensor networks," in *Algorithms—ESA 2006*, Y. Azar and T. Erlebach, Eds., vol. 4168, pp. 88–99, Springer, Berlin, Germany, 2006.
- [26] O. Chipara, C. Lu, and J. Stankovic, "Dynamic conflict-free query scheduling for wireless sensor networks," in *Proceedings* of the 14th IEEE International Conference on Network Protocols (ICNP '06), pp. 321–331, November 2006.
- [27] M. O. Díaz-Anadón and K. K. Leung, "TDMA scheduling for event-triggered data aggregation in irregular wireless sensor networks," *Computer Communications*, vol. 34, no. 17, pp. 2072– 2081, 2011.
- [28] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388– 404, 2000.

# Research Article **Doubly Constrained Robust Blind Beamforming Algorithm**

### Xin Song, Jingguo Ren, and Qiuming Li

Engineering Optimization and Smart Antenna Institute, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China

Correspondence should be addressed to Xin Song; songxin00@126.com

Received 28 December 2012; Accepted 15 July 2013

Academic Editor: Bin Wang

Copyright © 2013 Xin Song et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose doubly constrained robust least-squares constant modulus algorithm (LSCMA) to solve the problem of signal steering vector mismatches via the Bayesian method and worst-case performance optimization, which is based on the mismatches between the actual and presumed steering vectors. The weight vector is iteratively updated with penalty for the worst-case signal steering vector by the partial Taylor-series expansion and Lagrange multiplier method, in which the Lagrange multipliers can be optimally derived and incorporated at each step. A theoretical analysis for our proposed algorithm in terms of complexity cost, convergence performance, and SINR performance is presented in this paper. In contrast to the linearly constrained LSCMA, the proposed algorithm provides better robustness against the signal steering vector mismatches, yields higher signal captive performance, improves greater array output SINR, and has a lower computational cost. The simulation results confirm the superiority of the proposed algorithm on beampattern control and output SINR enhancement.

### 1. Introduction

Adaptive beamforming, as an attractive solution to signal detection and estimation in harsh environments, has received considerable attention in the fields of radar, sonar, seismology, radio astronomy, medical imaging, artificial intelligence, and neural network [1-5]. Many methods have been proposed and received great attention in the last twenty years. The class of blind adaptive beamforming algorithms is generally known as LSCMA, which can be rapidly convergent and globally stable for any linearly independent set of input signals. However, in the complex communication environments, adaptive beamforming algorithms may suffer significant performance degradation in the presence of the signal mismatches between the actual and assumed signal steering vectors. Such a type of mismatches may occur due to the unknown deformation of the antenna or sensor array, steer direction errors, and the drifting effect in the electronics or the multipath propagation, and so forth. Many solutions have been proposed such as convex quadratic constraints [6] and Bayesian approach [7] to account for the steering direction error of the target source. The eigenspace-based algorithm [8, 9] is also a good approach for robust beamforming. However, it is inefficient at low input signal-to-noise ratio (SNR) due to the substantial possibility of subspace swap. Recently, some advanced methods have been proposed [10–19]. It is proven that many robust adaptive beamformers belong to the family of the diagonal loading method. With the generalization of the sphere uncertainty set to ellipsoid, the optimal diagonal loading level can be calculated efficiently by the proposed methods [10, 11]. A very effective approach developed to the design of robust adaptive beamforming is based on the principle of worst-case performance optimization [12-14]. This approach delimits the uncertainty set of steering vectors by upper-bounding the norm of the difference between the actual and presumed steering vectors, that is, the norm of the mismatch vector. The value of the upper bound is assumed to be known. Regrettably, the second-order cone programming (SOCP) method does not provide a closed-form solution for the weight vector, and even it cannot be implemented online [15]. The general rank case has been considered, and an elegant closed-form solution has been obtained [16]. In a multiuser multipleinput single-output (MISO) cognitive radio network, the design of robust downlink beamforming is presented [17]. To reduce the impact of the channel state information (CSI) errors, two robust beamforming schemes are proposed in [18], which recover the large fraction of the SINR lost due to the channel estimation errors, but ultimately a large enough channel mismatch can eliminate the secrecy advantage of using artificial noise. To mitigate the detrimental effect of interferers, we extend the one-dimensional covariance fitting approach to multidimensional covariance fitting, modeling the source steering vector by means of uncertainty sets [19].

In this paper, robust LSCMA based on double constraints is proposed via the worst-case performance optimization. The quadratic constraint on the weight vector can improve robustness to the signal steering vector mismatches. In order for LSCMA to provide improved performance, the updating weight vector subject to the constraints of distortionless array response is derived by the partial Taylor-series expansion and Lagrange multiplier method, in which the multipliers can be optimally derived and incorporated at each step. The implementation of the proposed algorithm based on iterative minimization eliminates the covariance matrix inversion estimation, so it has a low computational load. Compared with the linearly constrained LSCMA, the proposed algorithm suffers the least distortion from the direction near the desired steering angle, yields better signal captive performance, and has superior performance on SINR improvement. The theoretical analysis and simulation results have been carried out to demonstrate effectiveness and superiority of the proposed algorithm in the signal steering vector mismatches. So the proposed algorithm can be an appealing technique and be implemented in digital system to improve the receiver performance.

### 2. Problem Formulation

2.1. Signal Model. We assume that there are M sensors and D unknown sources impinging from directions  $\{\theta_0, \theta_1, \dots, \theta_{D-1}\}$ . The sensors receive the linear combination of the source signals in the presence of additive white Gaussian noise (AWGN). Therefore, the received signal vector is given by

$$\mathbf{x} (k) = s_0 (k) \mathbf{a} (\theta_0) + \mathbf{i} (k) + \mathbf{n} (k)$$
  
=  $\mathbf{A}_D \mathbf{S} (k) + \mathbf{n} (k)$ , (1)

where  $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_M(k)]^T$  is the observed signal vector,  $\mathbf{a}(\theta_0)$  is the desired signal steering vector,  $\mathbf{A}_D = [\mathbf{a}(\theta_0), \mathbf{a}(\theta_1), \dots, \mathbf{a}(\theta_{D-1})]$  is the array manifold,  $\mathbf{S}(k)$ is the vector of *D* transmitted signals,  $\mathbf{i}(k)$  is the interference components, and  $\mathbf{n}(k)$  is the noise components with zero mean. The aim of blind adaptive beamforming is to estimate the source signal  $s_0(k)$  using only the observed data  $\mathbf{x}(k)$ . We write the estimated source signal as

$$y(k) = \mathbf{w}^{\mathsf{H}}\mathbf{x}(k), \qquad (2)$$

where  $\mathbf{w} = [w_1, w_2, \dots, w_M]^T$  is the complex weight vector and  $(\cdot)^{T}$  and  $(\cdot)^{H}$  stand for the transpose and Hermitian transpose, respectively. The signal-to-interference-plus-noise ratio (SINR) has the following form:

$$\mathrm{SINR} = \frac{\sigma_s^2 \left| \mathbf{w}^{\mathrm{H}} \mathbf{a} \left( \theta_0 \right) \right|^2}{\mathbf{w}^{\mathrm{H}} \mathbf{R}_{i+n} \mathbf{w}},$$
(3)

where  $\sigma_s^2$  is the signal power and  $\mathbf{R}_{i+n}$  is the  $M \times M$ interference-plus-noise covariance matrix:

$$\mathbf{R}_{i+n} = E\left\{ \left(\mathbf{i}\left(k\right) + \mathbf{n}\left(k\right)\right) \left(\mathbf{i}\left(k\right) + \mathbf{n}\left(k\right)\right)^{\mathrm{H}} \right\},\qquad(4)$$

where  $E[\cdot]$  denotes statistical expectation.

In the array signal processing, the objective of adaptive beamforming is to enhance the desired signal and suppress the noise and interference signals, which improves the array output SINR. In the adaptive array antenna system, the output SINR achieved the optimal one by regulating the weight vector.

2.2. The Linearly Constrained LSCMA. The linearly constrained LSCMA that is an effective solution to the problem of interference capture can be used for equalization, blind adaptive beamforming, and other similar applications when the desired signal has a constant envelope [20]. It is formulated as the following optimization problem:

min 
$$g(\mathbf{w}) = \sum_{i}^{K} |f_i(\mathbf{w})|^2 = \|\mathbf{f}(\mathbf{w})\|_2^2$$
 (5)  
subject to  $\mathbf{w}^{H} \mathbf{a}(\theta_0) = 1$ ,

where  $\mathbf{f}(\mathbf{w}) = [f_1(\mathbf{w}), \dots, f_K(\mathbf{w})]^T$  and  $f_i(\mathbf{w}) = |\mathbf{w}^H \mathbf{x}(i)| - 1$ . We define  $\mathbf{X}_K = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(K)]$  and solve (5) to

obtain the weight vector

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \widehat{\mathbf{R}}_{K}^{-1} \left[ \mathbf{X}_{K} \mathbf{e}^{\mathsf{H}} + \mathbf{a}(\theta_{0}) \delta \right], \qquad (6)$$

where

$$\widehat{\mathbf{R}}_{K} = \sum_{i=1}^{K} \mathbf{x} \left( i \right) \mathbf{x}^{\mathrm{H}} \left( i \right),$$

$$\delta = -\frac{1 - \mathbf{a}^{\mathrm{H}} \left( \theta_{0} \right) \mathbf{w} \left( k \right) + \mathbf{a}^{\mathrm{H}} \left( \theta_{0} \right) \widehat{\mathbf{R}}_{K}^{-1} \mathbf{X}_{K} \mathbf{e}^{\mathrm{H}}}{\mathbf{a}^{\mathrm{H}} \left( \theta_{0} \right) \widehat{\mathbf{R}}_{K}^{-1} \mathbf{a} \left( \theta_{0} \right)}, \qquad (7)$$

$$\mathbf{e} = \left[ e_{1}, \dots, e_{K} \right] = \left[ y_{1} - \frac{y_{1}}{|y_{1}|}, \dots, y_{K} - \frac{y_{K}}{|y_{K}|} \right].$$

From (6), we note that the constrained LSCMA requires the precise steering vector of the desired signal. But in practical applications, this may bring the mismatch between the presumed and actual signal steering vectors, because some of the underlying assumptions on the environments, sources, or sensor array can be violated. Therefore, the linearly constrained LSCMA is very sensitive to the signal steering vector mismatches, which causes serious cancellation problem of the desired signal.

### 3. Robust Constrained LSCMA under **Double Constraints**

To overcome the above-mentioned problem, robust constrained LSCMA is proposed, which provides excellent robustness against signal steering vector mismatches, suppresses the interference signals effectively, and enhances the array output SINR. In practical applications, the array beampattern error is formulated as

$$\vartheta^{2} = \mathbf{w}^{\mathrm{H}} \left[ \int_{\theta_{0} - \Delta \theta}^{\theta_{0} + \Delta \theta} \cos \theta \left( \mathbf{\tilde{a}} - \mathbf{\bar{a}} \right) \left( \mathbf{\tilde{a}} - \mathbf{\bar{a}} \right)^{\mathrm{H}} \mathrm{d}\theta \right] \mathbf{w}$$

$$= \mathbf{w}^{\mathrm{H}} \mathbf{Q} \mathbf{w},$$
(8)

where  $\mathbf{Q} = \int_{\theta_0 - \Delta \theta}^{\theta_0 + \Delta \theta} (\widetilde{\mathbf{a}} \widetilde{\mathbf{a}}^H - \widetilde{\mathbf{a}} \widetilde{\mathbf{a}}^H - \overline{\mathbf{a}} \widetilde{\mathbf{a}}^H + \overline{\mathbf{a}} \cdot \overline{\mathbf{a}}^H) d\sin \theta$ ,  $\overline{\mathbf{a}}$  is the assumed steering vector, and  $\widetilde{\mathbf{a}}$  is the estimated steering vector with mismatches.

The cost function of robust constrained LSCMA can be written as

$$\min_{\mathbf{w}} g(\mathbf{w}) = \sum_{i=1}^{K} \left\| \mathbf{w}^{\mathrm{H}} \mathbf{x}(i) \right\| - 1 \right\|^{2} \quad \text{subject to } \mathbf{w}^{\mathrm{H}} \mathbf{Q} \mathbf{w} \le \xi^{2},$$
(9)

where  $\xi$  is a given integer. The new formulation (9) is based on the worst-case performance optimization because it implies that  $g(\mathbf{w})$  is minimized subject to the constraint and the distortionless array response will be maintained for the worst-case mismatch  $\Delta \theta$ .

The quadratic constraint is adjoined to the cost function by the Lagrange multiplier  $\eta$ , and we can obtain the Lagrange function  $H(\mathbf{w}, \eta)$ :

$$H\left(\mathbf{w},\eta\right) = \frac{1}{2}\sum_{i=1}^{K} \left\| \mathbf{w}^{\mathsf{H}}\mathbf{x}\left(i\right) \right\| - 1 \right\|^{2} + \frac{1}{2}\eta\left(\mathbf{w}^{\mathsf{H}}\mathbf{Q}\mathbf{w} - \xi^{2}\right).$$
(10)

The partial Taylor-series expansion of (10) is

$$H\left(\mathbf{w} + \mathbf{d}, \eta\right) = \frac{1}{2} \left\| \mathbf{f}\left(\mathbf{w}\right) + \mathbf{J}^{\mathrm{H}}\left(\mathbf{w}\right) \mathbf{d} \right\|_{2}^{2} + \frac{1}{2} \eta \left(\mathbf{w}^{\mathrm{H}} \mathbf{Q} \mathbf{w} - \xi^{2}\right) + \eta \mathbf{w}^{\mathrm{H}} \mathbf{Q}^{\mathrm{H}} \mathbf{d},$$
(11)

where **d** is the offset vector and J(w) is the Jacobian of f(w):

$$\mathbf{J}(\mathbf{w}) = \nabla_{\mathbf{w}} \mathbf{f}(\mathbf{w}) = \left[\nabla_{\mathbf{w}} f_{1}(\mathbf{w}), \dots, \nabla_{\mathbf{w}} f_{K}(\mathbf{w})\right],$$
  
$$\nabla_{\mathbf{w}} f_{i}(\mathbf{w}) = \nabla_{\mathbf{w}} \left\{ \left| \mathbf{w}^{\mathrm{H}} \mathbf{x}(i) \right| - 1 \right\} = \mathbf{x}(i) \frac{y_{i}^{*}}{|y_{i}|}.$$
(12)

Take the gradient of  $H(\mathbf{w} + \mathbf{d}, \eta)$  with respect to  $\mathbf{d}$ :

$$\nabla_{\mathbf{d}} \left( H \left( \mathbf{w} + \mathbf{d}, \eta \right) \right) = \mathbf{J} \left( \mathbf{w} \right) \mathbf{J}^{\mathrm{H}} \left( \mathbf{w} \right) \mathbf{d} + \mathbf{J} \left( \mathbf{w} \right) \mathbf{f} \left( \mathbf{w} \right) + \eta \mathbf{Q} \mathbf{w}.$$
(13)

By equating (13) to zero, the offset vector **d** can be calculated as

$$\mathbf{d} = -\left[\mathbf{J}\left(\mathbf{w}\right)\mathbf{J}^{\mathrm{H}}\left(\mathbf{w}\right)\right]^{-1}\left[\mathbf{J}\left(\mathbf{w}\right)\mathbf{f}\left(\mathbf{w}\right) + \eta\mathbf{Q}\mathbf{w}\right].$$
(14)

Using (12), we can derive the following equation simply:

$$\mathbf{J}(\mathbf{w}) \mathbf{J}^{\mathrm{H}}(\mathbf{w}) = \mathbf{X}_{K} \mathbf{X}_{K}^{\mathrm{H}} = \widehat{\mathbf{R}}_{K},$$

$$\mathbf{J}(\mathbf{w}) \mathbf{f}(\mathbf{w}) = \mathbf{X}_{K} \mathbf{e}^{\mathrm{H}}.$$
(15)

Combining (15) and (14), we can rewrite the offset vector

$$\mathbf{d} = -\widehat{\mathbf{R}}_{K}^{-1} \left[ \mathbf{X}_{K} \mathbf{e}^{\mathrm{H}} + \eta \mathbf{Q} \mathbf{w} \right].$$
(16)

Then, using (16), the updating weight vector for robust constrained LSCMA becomes

$$\mathbf{w} (k+1) = \mathbf{w} (k) - \widehat{\mathbf{R}}_{K}^{-1} \left[ \mathbf{X}_{K} \mathbf{e}^{\mathrm{H}} + \eta \mathbf{Q} \mathbf{w} (k) \right]$$
  
=  $\left( \mathbf{I} - \eta \widehat{\mathbf{R}}_{K}^{-1} \mathbf{Q} \right) \mathbf{w} (k) - \widehat{\mathbf{R}}_{K}^{-1} \mathbf{X}_{K} \mathbf{e}^{\mathrm{H}}.$  (17)

From (6) and (17), we remark that the major computational demand to derive the weight vector comes from the covariance matrix inversion, which requires  $O(M^3)$  flops. This leads to a high computational cost. In our proposed algorithm, to solve this problem, the iterative method is used to calculate the covariance matrix inversion. Using the matrix inverse lemma, we can obtain

$$\mathbf{G}(K) = \mathbf{R}_{K}^{-1}$$

$$= \left[ \mathbf{G}(K-1) - \frac{\mathbf{G}(K-1)\mathbf{x}(K)\mathbf{x}^{\mathrm{H}}(K)\mathbf{G}(K-1)}{1 + \mathbf{x}^{\mathrm{H}}(K)\mathbf{G}(K-1)\mathbf{x}(K)} \right].$$
(18)

Inserting (18) into (17), the weight vector is updated as

$$\mathbf{w}(k+1) = \left(\mathbf{I} - \eta \mathbf{G}(k) \mathbf{Q}\right) \mathbf{w}(k) - \mathbf{G}(k) \mathbf{X}_{K} \mathbf{e}^{\mathrm{H}}.$$
 (19)

Equation (19) is substituted into the quadratic constraint in (9), which yields

$$\left(\mathbf{F}(k) - \eta \mathbf{D}(k)\right)^{\mathrm{H}} \mathbf{Q}\left(\mathbf{F}(k) - \eta \mathbf{D}(k)\right) = \xi^{2}, \qquad (20)$$

where

$$\mathbf{F}(k) = \mathbf{w}(k) - \mathbf{G}(k) \mathbf{X}_{K} \mathbf{e}^{H},$$
  
$$\mathbf{D}(k) = \mathbf{G}(k) \mathbf{Q} \mathbf{w}(k).$$
(21)

To solve (20), the Lagrange multiplier  $\eta$  has the following form:

$$\eta = \frac{\operatorname{Re}\left[\mathbf{F}^{\mathrm{H}}\left(k\right)\mathbf{QD}\left(k\right)\right] - \operatorname{Re}\left[\rho\left(k\right)\right]}{\mathbf{D}^{\mathrm{H}}\left(k\right)\mathbf{QD}\left(k\right)},$$
(22)

where

$$\rho^{*}(k) \rho(k) = \left( \operatorname{Re} \left[ \mathbf{F}^{\mathrm{H}}(k) \mathbf{Q} \mathbf{D}(k) \right] \right)^{2} - \left[ \mathbf{D}^{\mathrm{H}}(k) \mathbf{Q} \mathbf{D}(k) \left( \mathbf{F}^{\mathrm{H}}(k) \mathbf{Q} \mathbf{F}(k) - \xi^{2} \right) \right].$$
(23)

In order to detect the desired signal under directional uncertainty, we can impose another constraint on an average steering vector via the Bayesian approach. We assume that the direction of arrival (DOA) is a discrete random variable with known a priori probability density function (pdf)  $q(\tilde{\theta})$  that reflects the level of uncertainty about the source DOA. For computational simplicity, we assume that  $q(\tilde{\theta})$  is defined only

on a discrete set of *L* points,  $\Theta = {\tilde{\theta}_1, \tilde{\theta}_2, ..., \tilde{\theta}_L}$ , in the a priori parameter space.

When interferers are present, the a posteriori probability density function  $p(\tilde{\theta}_i | \mathbf{X}_K)$  is difficult to implement because it is a function of  $\mathbf{R}_{i+n}$ , which is unknown and hard to estimate. We derive approximate  $p(\tilde{\theta}_i | \mathbf{X}_K)$  with a simpler expression [7]

$$\widehat{p}\left(\widetilde{\theta}_{i} \mid \mathbf{X}_{K}\right) = \frac{q\left(\widetilde{\theta}_{i}\right)\exp\left\{\beta K\left(\mathbf{a}^{\mathrm{H}}\left(\widetilde{\theta}_{i}\right)\widehat{\mathbf{R}}_{K}^{-1}\mathbf{a}\left(\widetilde{\theta}_{i}\right)\right)^{-1}\right\}}{\sum_{j=1}^{L}q\left(\widetilde{\theta}_{j}\right)\exp\left\{\beta K\left(\mathbf{a}^{\mathrm{H}}\left(\widetilde{\theta}_{j}\right)\widehat{\mathbf{R}}_{K}^{-1}\mathbf{a}\left(\widetilde{\theta}_{j}\right)\right)^{-1}\right\}},$$
(24)

where  $\beta$  is a monotonically increasing function of SNR.

At low SNR, it will be relatively flat over all DOAs and revert to the a priori pdf. At high SNR, the a posteriori probability of the true DOA will approach one, whereas the posteriori probability of the other DOAs will approach zero.

Based on (24), we can consider that a further form of robust constrained LSCMA (9) is expressed as

$$\min_{\mathbf{w}} \sum_{i=1}^{K} \left\| \mathbf{w}^{\mathrm{H}} \mathbf{x}(i) \right\| - 1 \right\|^{2} \quad \text{subject to } \mathbf{w}^{\mathrm{H}} \widehat{\mathbf{a}}_{\mathrm{v}} = 1, \ \mathbf{w}^{\mathrm{H}} \mathbf{Q} \mathbf{w} \le \xi^{2},$$
(25)

where  $\hat{\mathbf{a}}_{v}$  is an average steering vector averaged over  $\hat{p}(\bar{\theta}_{i} | \mathbf{X}_{K})$ :

$$\widehat{\mathbf{a}}_{\mathrm{v}} = \sum_{i=1}^{L} \mathbf{a}\left(\widetilde{\theta}_{i}\right) \widehat{p}\left(\widetilde{\theta}_{i} \mid \mathbf{X}_{K}\right) = \mathbf{A}\mathbf{p}, \qquad (26)$$

where  $\mathbf{A} = [\mathbf{a}(\tilde{\theta}_1), \mathbf{a}(\tilde{\theta}_2), \dots, \mathbf{a}(\tilde{\theta}_L)]$  is the  $M \times L$  matrix of steering vectors and  $\mathbf{p}$  is the  $L \times 1$  vector.

Gauss's method updates w by the offset d that minimizes the partial Taylor-series expansion of (25) subject to the double constraints; that is,

$$\min_{\mathbf{w}} g\left(\mathbf{w} + \widehat{\mathbf{d}}\right) \approx \left\| \mathbf{f}\left(\mathbf{w}\right) + \mathbf{J}^{\mathrm{H}}\left(\mathbf{w}\right) \widehat{\mathbf{d}} \right\|_{2}^{2}$$
subject to  $\mathbf{w}^{\mathrm{H}} \widehat{\mathbf{a}}_{\mathrm{v}} = 1, \ \mathbf{w}^{\mathrm{H}} \mathbf{Q} \mathbf{w} \leq \xi^{2}.$ 

$$(27)$$

Using the Lagrange multiplier method, the optimal solution to (27) is obtained by minimizing the following function:

$$\begin{split} \widehat{H}\left(\mathbf{w}+\widehat{\mathbf{d}},\lambda,\gamma\right) &= \frac{1}{2}\left\|\mathbf{f}\left(\mathbf{w}\right)+\mathbf{J}^{\mathrm{H}}\left(\mathbf{w}\right)\widehat{\mathbf{d}}\right\|_{2}^{2}+\lambda\left(\mathbf{w}^{\mathrm{H}}\widehat{\mathbf{a}}_{\mathrm{v}}-1\right)\right.\\ &+ \frac{1}{2}\gamma\left(\mathbf{w}^{\mathrm{H}}\mathbf{Q}\mathbf{w}-\xi^{2}\right)+\lambda\widehat{\mathbf{a}}_{\mathrm{v}}^{\mathrm{H}}\widehat{\mathbf{d}}+\gamma\mathbf{w}^{\mathrm{H}}\mathbf{Q}^{\mathrm{H}}\widehat{\mathbf{d}}, \end{split}$$
(28)

where  $\lambda$  and  $\gamma$  are the Lagrange multipliers.

The gradient of  $H(\mathbf{w} + \mathbf{d}, \lambda, \gamma)$  with respect to  $\mathbf{d}$  is

$$\nabla_{\widehat{\mathbf{d}}} \left( \widehat{H} \left( \mathbf{w} + \widehat{\mathbf{d}}, \lambda, \gamma \right) \right)$$
  
=  $\frac{1}{2} \nabla_{\widehat{\mathbf{d}}} \left\{ \left[ \mathbf{f} \left( \mathbf{w} \right) + \mathbf{J}^{\mathrm{H}} \left( \mathbf{w} \right) \widehat{\mathbf{d}} \right]^{\mathrm{H}} \left[ \mathbf{f} \left( \mathbf{w} \right) + \mathbf{J}^{\mathrm{H}} \left( \mathbf{w} \right) \widehat{\mathbf{d}} \right] \right\}$ (29)  
+  $\lambda \widehat{\mathbf{a}}_{\mathrm{v}} + \gamma \mathbf{Q} \mathbf{w}$   
=  $\mathbf{J} \left( \mathbf{w} \right) \mathbf{f} \left( \mathbf{w} \right) + \mathbf{J} \left( \mathbf{w} \right) \mathbf{J}^{\mathrm{H}} \left( \mathbf{w} \right) \widehat{\mathbf{d}} + \lambda \widehat{\mathbf{a}}_{\mathrm{v}} + \gamma \mathbf{Q} \mathbf{w}.$ 

We can obtain the offset vector  $\mathbf{d}$ :

$$\widehat{\mathbf{d}} = -\left[\mathbf{J}\left(\mathbf{w}\right)\mathbf{J}^{\mathrm{H}}\left(\mathbf{w}\right)\right]^{-1}\left[\mathbf{J}\left(\mathbf{w}\right)\mathbf{f}\left(\mathbf{w}\right) + \lambda\widehat{\mathbf{a}}_{\mathrm{v}} + \gamma \mathbf{Q}\mathbf{w}\right].$$
(30)

Then, the updating weight vector for robust constrained LSCMA becomes

$$\mathbf{w} (k+1) = \mathbf{w} (k) - \mathbf{G} (k) \left[ \mathbf{X}_{K} \mathbf{e}^{H} + \hat{\mathbf{a}}_{v} \lambda + \gamma \mathbf{Q} \mathbf{w} (k) \right]$$
$$= \left( \mathbf{I} - \gamma \mathbf{G} (k) \mathbf{Q} \right) \mathbf{w} (k) - \mathbf{G} (k) \mathbf{X}_{K} \mathbf{e}^{H} - \lambda \mathbf{G} (k) \hat{\mathbf{a}}_{v}.$$
(31)

Substituting (31) into the linear constraint of (27), we can get

$$\lambda = \alpha \left( \mathbf{p}^{\mathrm{T}} \mathbf{A}^{\mathrm{H}} \mathbf{w} \left( k \right) - \mathbf{p}^{\mathrm{T}} \mathbf{A}^{\mathrm{H}} \mathbf{G} \left( k \right) \mathbf{X}_{K} \mathbf{e}^{\mathrm{H}} -\gamma \mathbf{p}^{\mathrm{T}} \mathbf{A}^{\mathrm{H}} \mathbf{G} \left( k \right) \mathbf{Q} \mathbf{w} \left( k \right) - 1 \right),$$
(32)

where  $\alpha = (\mathbf{p}^{\mathrm{T}} \mathbf{A}^{\mathrm{H}} \mathbf{G}(k) \mathbf{A} \mathbf{p})^{-1}$ .

By inserting the multiplier  $\lambda$  into (31), the weight vector can be rewritten as

$$\mathbf{w} (k + 1)$$

$$= \mathbf{w} (k) - \mathbf{G} (k) \mathbf{X}_{K} \mathbf{e}^{H} - \alpha \mathbf{G} (k) \mathbf{p}^{T} \mathbf{A}^{H} \mathbf{w} (k) \mathbf{A} \mathbf{p}$$

$$+ \alpha \mathbf{G} (k) \mathbf{p}^{T} \mathbf{A}^{H} \mathbf{G} (k) \mathbf{X}_{K} \mathbf{e}^{H} \mathbf{A} \mathbf{p} + \alpha \mathbf{G} (k) \mathbf{A} \mathbf{p}$$

$$- \gamma \left[ \mathbf{G} (k) \mathbf{Q} \mathbf{w} (k) - \alpha \mathbf{G} (k) \mathbf{p}^{T} \mathbf{A}^{H} \mathbf{G} (k) \mathbf{Q} \mathbf{w} (k) \mathbf{A} \mathbf{p} \right].$$
(33)

Inserting (33) into the quadratic constraint of (27), we can obtain

$$\gamma = \frac{\operatorname{Re}\left[\mathbf{P}^{\mathrm{H}}\left(k\right)\mathbf{QV}\left(k\right)\right] - \operatorname{Re}\left[\chi\left(k\right)\right]}{\mathbf{V}^{\mathrm{H}}\left(k\right)\mathbf{QV}\left(k\right)},$$
(34)

where

$$\mathbf{P}(k) = \mathbf{w}(k) - \mathbf{G}(k) \mathbf{X}_{K} \mathbf{e}^{H} - \alpha \mathbf{G}(k) \mathbf{p}^{T} \mathbf{A}^{H} \mathbf{w}(k) \mathbf{A} \mathbf{p}$$

$$+ \alpha \mathbf{G}(k) \mathbf{p}^{T} \mathbf{A}^{H} \mathbf{G}(k) \mathbf{X}_{K} \mathbf{e}^{H} \mathbf{A} \mathbf{p} + \alpha \mathbf{G}(k) \mathbf{A} \mathbf{p},$$

$$\mathbf{V}(k) = \mathbf{G}(k) \mathbf{Q} \mathbf{w}(k) - \alpha \mathbf{G}(k) \mathbf{p}^{T} \mathbf{A}^{H} \mathbf{G}(k) \mathbf{Q} \mathbf{w}(k) \mathbf{A} \mathbf{p},$$

$$\chi^{*}(k) \chi(k) = \left( \operatorname{Re} \left[ \mathbf{P}^{H}(k) \mathbf{Q} \mathbf{V}(k) \right] \right)^{2}$$

$$- \left[ \mathbf{V}^{H}(k) \mathbf{Q} \mathbf{V}(k) \left( \mathbf{P}^{H}(k) \mathbf{Q} \mathbf{P}(k) - \xi \right) \right].$$
(36)

### 4. Performance Analysis

4.1. *The Implementation Complexity Cost.* The complexity cost of the conventional LSCMA and the proposed robust LSCMA can be shown in Tables 1 and 2.

4.2. Convergence Performance. The proposed robust constrained LSCMA is globally stable and convergent via Agee's inequalities. The first input stream is successfully extracted by establishing the following inequalities, given that i > 0 [21]:

$$0 \le \|\mathbf{y}_{1}(i)\|_{2}^{2} \le \|\mathbf{y}_{1}(i+1)\|_{2}^{2} \le K,$$
  

$$0 \le \|\mathbf{y}_{1}(i+1) - \mathbf{y}_{1}(i)\|_{2}^{2} \le g(\mathbf{w}_{1}(i)) \le \|\mathbf{y}_{1}(i+1)\|_{2}^{2} \le K,$$
  

$$0 \le g(\mathbf{w}_{1}(i+1)) \le g(\mathbf{w}_{1}(i)) - \|\mathbf{y}_{1}(i+1) - \mathbf{y}_{1}(i)\|_{2}^{2} \le K,$$
  
(37)

where  $\mathbf{y}_1(i) = \mathbf{w}_1^{\mathrm{H}}(i)\mathbf{X}_1(i)$  and  $\mathbf{X}_1(i) = \mathbf{X}_K$ .

To extract the second input stream, we begin with the convergence of  $\mathbf{X}_2(i)$  and then we have  $\mathbf{y}_2(i) = \mathbf{w}_2^{\mathrm{H}}(i)\mathbf{X}_2(i)$ , which is convergent via

$$0 \le \|\mathbf{y}_{2}(i)\|_{2}^{2} \le \|\mathbf{y}_{2}(i+1)\|_{2}^{2} \le K.$$
(38)

Then, each output stream is convergent via

$$0 \le \left\| \mathbf{y}_{m}(i) \right\|_{2}^{2} \le \left\| \mathbf{y}_{m}(i+1) \right\|_{2}^{2} \le K.$$
(39)

For i > 0, the overall performance may either improve or be maintained; thus the convergence performance of the proposed LSCMA is established.

4.3. Output SINR Performance. The output signal of the proposed beamformer can be expressed as

$$y(k) = \mathbf{w}^{\mathrm{H}}(k) \mathbf{x}(k)$$
  
=  $e^{j\varphi(k)} + \omega m(k) e^{j\phi(k)}$ , (40)

where  $\varphi(k)$  is the phase of the desired signal  $s_0(k)$ ,  $\omega$  controls the SINR, and m(k) and  $\varphi(k)$  are the magnitude and phase of the interference term, respectively. The normalized output is

$$l(k) = \frac{y(k)}{|y(k)|} = \tau s_0(k) + \rho z(k) + v(k), \qquad (41)$$

where the parameters  $\tau$ ,  $\rho$  determine the power of the desired signal and the interference and v(k) contains the intermodulation terms. Here we are interested only in the power of the desired signal and the interference in the normalized output, and the intermodulation terms are ignored.

Assuming that the initial beamformer SINR is known, the SINR of the normalized output signal l(k) can be calculated [22]:

SINR = 
$$\frac{\tau^2}{\rho^2} = \frac{\left|E\left[s_0\left(k\right)l^*\left(k\right)\right]\right|^2}{\left|E\left[z\left(k\right)l^*\left(k\right)\right]\right|^2}.$$
 (42)

	The complexity cost
$\widehat{\mathbf{R}}_{K}$	$O(M^2 \times K)$
$\widehat{\mathbf{R}}_{K}^{-1}$	$O(M^3)$
δ	$O(2M^2 + M \times K + 2M + K + 1)$
$\widehat{\mathbf{R}}_{K}^{-1}\mathbf{X}_{K}\mathbf{e}^{\mathrm{H}}$	$O(M^2 \times K + M \times K)$
$\widehat{\mathbf{R}}_{K}^{-1}\mathbf{a}(\mathbf{ heta}_{0})\mathbf{\delta}$	$O(M^2 + M)$
Total complexity cost	$O(M^3 + (2K+3) \times M^2 + (2K+3) \times M + K + 1)$

TABLE 2: The complexity cost of the proposed LSCMA.

	The complexity cost
<b>Q</b> ( <i>k</i> )	$O(4M^2)$
$\mathbf{G}(k)$	$O(3M^2 + 2M)$
Ар	$O(M \times L)$
d	$O(2M^2 + (K + L + 2) \times M)$
λ	$O(M^2 + 4M + 2)$
$\mathbf{P}(k)$	$O(4M^2 + (K+1) \times M + K)$
$\mathbf{V}(k)$	$O(3M^2 + M)$
$\chi(k)$	$O(3M^2 + 3M + 1)$
Total complexity cost	$O(20M^2 + (2K + 2L + 13) \times M + K + 3)$

We assume  $\omega \ll 1$ , so the cross-correlation of  $s_0(k)$  and l(k) is

$$\tau = E\left[\left(1 + \omega m\left(k\right)e^{j\left(\varphi(k) - \phi(k)\right)}\right) \times \left(1 - \omega m\left(k\right)\cos\left(\varphi\left(k\right) - \phi\left(k\right)\right)\right)\right] \simeq 1.$$
(43)

Similarly the cross-correlation of z(k) and l(k) is

$$\rho = E\left[\left(m\left(k\right)e^{j\left(\phi\left(k\right)-\varphi\left(k\right)\right)} + \omega m^{2}\left(k\right)\right) \times \left(1 - \omega m\left(k\right)\cos\left(\varphi\left(k\right) - \phi\left(k\right)\right)\right)\right]$$
(44)
$$= \frac{\omega}{2}.$$

Inserting (43) and (44) to (42), we can obtain the output SINR as  $4/\omega^2$ , so the normalized output of the proposed algorithm increases the array output SINR.

#### **5. Simulation Results**

In this section, Matlab software is used to evaluate the performance of the proposed algorithm. The sampling frequency is  $F = 3f_s$  in the narrowband signal processing, where  $f_s$  is the maximal signal frequency. The sine wave signal source has been used for simulations. The block diagram to clarify the simulation scenario is shown in Figure 1. A uniform linear array of M = 10 sensors spaced half-wavelength apart is considered. All results are obtained by averaging 100 independent simulation runs. In all experiments, the nondirectional noise is assumed to be a spatially white Gaussian noise with unit covariance. It is assumed that there



FIGURE 1: The simulation scenario diagram.

is one desired source at 3° and two interfering sources at  $-50^{\circ}$  and 50°. The a priori uncertainty in the DOA is over the region  $u = \sin(\theta) \in [-0.2, 0.2]$ . The set  $\Theta$  is composed of L = 20 evenly spaced points on the interval [-0.2, 0.2]. First, we show the performance of array beampattern. Next, we investigate the performance of SINR improvement brought by the proposed method. The constrained parameter  $\xi^2 = 0.03$  is chosen for robust constrained LSCMA.

Example 1 (array beampatterns of two algorithms). The SNR is equal to 10 dB. The aforementioned algorithms are simulated by using a mismatched steering vector of the desired signal, where the practical angle of incidence equals 6°. This corresponds to a  $\Delta = 3^{\circ}$  mismatch in the signal look direction. Figure 2 shows the array beampatterns of the above-mentioned algorithms against snapshots for the no mismatch case. The vertical line in the figure represents direction of arrival of the desired signal. It is clear that in the two algorithms deep nulls are formed at the directions of interferences and the array has magnitude response. The mismatch case is illustrated in Figure 3, where the vertical line represents the direction of the actual signal. We find out that at a small mismatch the linearly constrained LSCMA treats the desired signal as a main beam interferer and is trying to suppress it, which leads to performance degradation. However, the proposed algorithm is better at providing robustness and having resolution compared with the linearly constrained LSCMA.

*Example 2* (output SINR versus snapshots). In the second experiment, the SINR performance of the aforesaid algorithms for the fixed SNR = 0 dB against the array imperfections is illustrated in Figures 4 and 5. In this example, the constrained LSCMA is very sensitive even to slight mismatches, which can cause signal cancellation problem. The result in Figure 5 shows that the proposed method offers about 5 dB improvement over the linearly constrained LSCMA and makes output SINR close to the optimal one due to the efficient handling of the average steering vector and worst-case performance optimization. It is clear that the proposed algorithm has superior performance on SINR improvement for no array imperfections.



FIGURE 2: Array beampattern (in no mismatch case).



FIGURE 3: Array beampattern (in the mismatch case).



FIGURE 4: Output SINR versus N (in no mismatch case).



FIGURE 5: Output SINR versus N (in the mismatch case).

*Example 3* (output SINR versus input SNR). In this experiment, we evaluate the SINR performance versus input SNR with DOA error for the fixed sample data size N = 100. The simulation results in Figures 6 and 7 indicate that the proposed method has slight performance degradation with the increasing of input SNR, and it is not sensitive to the power of the target signal. In this example, the performance of the linearly constrained LSCMA degrades when the signal power increases. Robust constrained LSCMA is known theoretically not to reach the optimal performance, but it is seen that for positive SNR the performance is almost identical to the optimal SINR. As expected, the sensitivity



FIGURE 6: Output SINR versus SNR (in no mismatch case).



FIGURE 7: Output SINR versus SNR (in the mismatch case).

to signal steering vector mismatches can be significantly lowered by the proposed algorithm. As a result, the proposed algorithm can provide sufficient robustness to pointing errors in perturbation situations.

### 6. Conclusions

In this paper, a novel robust LSCMA algorithm based on double constraints is proposed via the Bayesian approach and worst-case performance optimization. To improve robustness, the weight vector is optimized to involve minimization of the objective function with penalty for the worst-case signal steering vector by the partial Taylor-series expansion and Lagrange multiplier method, in which the parameters can be precisely derived at each iterative step. Moreover, the online implementation of the proposed algorithm eliminates the covariance matrix inversion estimation, which has a low computational load. The proposed robust constrained LSCMA has a faster convergence rate, provides better robustness against the signal steering vector mismatches, and yields improved array output performance compared with the linearly constrained LSCMA. The theoretical analysis and simulation experiments have been carried out to illustrate the significant performance improvement of the proposed method for the signal steering vector mismatches.

### Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments that helped improve the quality of this paper. This work was supported by the Program for New Century Excellent Talents in University no. NCET-12-0103, the Natural Science Foundation of Liaoning province under Grant no. 201102057, and the Natural Science Foundation of Hebei province under Grant no. F2012501044.

### References

- X. Huang, H.-C. Wu, and J. C. Principe, "Robust blind beamforming algorithm using joint multiple matrix diagonalization," *IEEE Sensors Journal*, vol. 7, no. 1, pp. 130–135, 2007.
- [2] S. A. Farès, T. A. Denidni, S. Affès, and C. Despins, "Fractionaldelay sequential blind beamforming for wireless multipath communications in confined areas," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 629–638, 2008.
- [3] J. Yang, H. Xi, F. Yang, and Y. Zhao, "Fast adaptive blind beamforming algorithm for antenna array in CDMA systems," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 2, pp. 549–558, 2006.
- [4] N. Xie, Y. Zhou, M. Xia, and W. Tang, "Fast blind adaptive beamforming algorithm with interference suppression," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1985–1988, 2008.
- [5] X. Song, J. Wang, and B. Wang, "Robust adaptive beamforming under quadratic constraint with recursive method implementation," *Wireless Personal Communications*, vol. 53, no. 4, pp. 555– 568, 2010.
- [6] B. D. van Veen, "Minimum variance beamforming with soft response constraints," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 1964–1972, 1991.
- [7] K. L. Bell, Y. Ephraim, and H. L. van Trees, "A Bayesian approach to robust adaptive beamforming," *IEEE Transactions on Signal Processing*, vol. 48, no. 2, pp. 386–398, 2000.
- [8] D. D. Feldman and L. J. Griffiths, "Projection approach for robust adaptive beamforming," *IEEE Transactions on Signal Processing*, vol. 42, no. 4, pp. 867–876, 1994.
- [9] L. Chang and C.-C. Yeh, "Performance of DMI and eigenspacebased beamformers," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 11, pp. 1336–1348, 1992.
- [10] J. Yang, X. Ma, C. Hou, and Y. Liu, "Automatic generalized loading for robust adaptive beamforming," *IEEE Signal Processing Letters*, vol. 16, no. 3, pp. 219–222, 2009.

- [11] J. Li, P. Stoica, and Z. Wang, "Doubly constrained robust Capon beamformer," *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2407–2423, 2004.
- [12] S. A. Vorobyov, A. B. Gershman, and Z.-Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: a solution to the signal mismatch problem," *IEEE Transactions* on Signal Processing, vol. 51, no. 2, pp. 313–324, 2003.
- [13] A. Elnashar, S. M. Elnoubi, and H. A. El-Mikati, "Further study on robust adaptive beamforming with optimum diagonal loading," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 12, pp. 3647–3658, 2006.
- [14] A. Gershman, N. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 62–75, 2010.
- [15] A. Elnashar, "Efficient implementation of robust adaptive beamforming based on worst-case performance optimisation," *IET Signal Processing*, vol. 2, no. 4, pp. 381–393, 2008.
- [16] S. Shahbazpanahi, A. B. Gershman, Z.-Q. Lou, and K. M. Wong, "Robust adaptive beamforming for general-rank signal models," *IEEE Transactions on Signal Processing*, vol. 51, no. 9, pp. 2257– 2269, 2003.
- [17] E. A. Gharavol, Y.-C. Liang, and K. Mouthaan, "Robust downlink beamforming in multiuser MISO cognitive radio networks with imperfect channel-state information," *IEEE Transactions* on Vehicular Technology, vol. 59, no. 6, pp. 2852–2860, 2010.
- [18] A. Mukherjee and A. L. Swindlehurst, "Robust beamforming for security in MIMO wiretap channels with imperfect CSI," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 351–361, 2011.
- [19] M. Rübsamen and A. B. Gershman, "Robust adaptive beamforming using multidimensional covariance fitting," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 740–753, 2012.
- [20] B. G. Agee, "The least-squares CMA: a new technique for rapid correction of constant modulus signal," in *Proceedings* of Acoustics, Speech and Signal Processing, pp. 953–956, Tokyo, Japan, April 1986.
- [21] P. Sansrimahachai, D. B. Ward, and A. G. Constantinides, "Blind source separation of instantaneous MIMO systems based on the least-squares constant modulus algorithm," *IEE Vision, Image and Signal Processing*, vol. 152, pp. 616–622, 2005.
- [22] T. E. Biedka, W. H. Tranter, and J. H. Reed, "Convergence analysis of the least squares constant modulus algorithm in interference cancellation applications," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 491–501, 2000.

### Research Article

## Adaptive Waveform Design for Multiple Radar Tasks Based on Constant Modulus Constraint

### Wang Bin, Yang Wenfang, and Wang Jinkuan

Northeastern University at Qinhuangdao, Qinhuangdao, Hebei 066004, China

Correspondence should be addressed to Wang Bin; wangbinneu@163.com

Received 29 May 2013; Accepted 12 July 2013

Academic Editor: Neal N. Xiong

Copyright © 2013 Wang Bin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cognitive radar is an intelligent system, and it can adaptively transmit waveforms to the complex environment. The intelligent radar system should be able to provide different trade-offs among a variety of performance objectives. In this paper, we investigate the mutual information (MI) in signal-dependent interference and channel noise. We propose a waveform design method which can efficiently synthesize waveforms and provide a trade-off between estimation performance and detection performance. After obtaining a local optimal waveform, we apply the technique of generating a constant modulus signal with the given Fourier transform magnitude to the waveform. Finally we obtain a waveform that has constant modulus property.

### 1. Introduction

Cognitive radar (CR) is a new concept of radar system proposed by Haykin in 2006 [1, 2]. In CR, the radar continuously learns about the environment through experience gained from interactions of the receiver with the environment, the transmitter adjusts its illumination of the environment in an intelligent manner and the whole radar system constitutes a closed-loop dynamic system. Therefore, adaptive waveform design is important to the performance of radar system. Recently, advances in flexible waveform generators and highspeed signal processing hardware have made it possible for transmitted waveforms to vary with the complex environment.

Many researches focused on waveform design for different tasks, for example, target detection, estimation, tracking, and recognition. An early attempt to the problem of matched waveform design for detecting a known target in additive Gaussian noise was addressed via the SNR criterion in [3]. From the frequency domain approach, the SNR-based optimal matched waveform for a known target in signal-dependent interference was derived in [4]. Information theory is also an important tool for waveform design. Bell [5] firstly proposed the method of maximizing the mutual information between the received signal and target impulse response to optimize the waveform, and many articles also used mutual information as the optimal criterion for waveform design [6, 7]. Because a more flexible design framework is required, CR should be able to provide different trade-offs among a variety of performance objectives. Haykin et al. [8] proposed a waveform design method that efficiently synthesizes waveforms which provide a trade-off between estimation performance for a Gaussian ensemble of targets and detection performance for a specific target in channel-noise-only environment.

In this paper, we will consider a situation when the signal-dependent interference is not negligible, and provide an optimal trade-off between the detection and estimation criteria. Thus we seek to maximize the mutual information between a random target impulse response and the received radar waveform, subjected to a lower bound on the SINR for the target and energy constraints. We assume that the target hypotheses are statistically characterized by known power spectral density (PSD) as in [9]. Therefore, the actual target realization is an unknown sample function generated from the PSD of the true target class.

One consideration in forming practical radar waveforms is the constant modulus constraint, which permits efficient use of the front-end power amplifier [10]. With proper manipulation of the waveforms in the temporal domain, it should be possible to design constant modulus waveforms that approximate MI-based waveform spectrum with some loss of optimality. Pillai et al. give us a technique of generating a constant modulus signal with the given Fourier transform magnitude in [11]; thus we can use this method to get a waveform that has constant modulus property.

This paper is organized in the following manner. Section 2 describes the target model for waveform design in signaldependent interference. Section 3 explains how to generate the constant modulus waveform from a given Fourier transform magnitude. Section 4 shows the derivation of the mutual information between the random target impulse response and the received radar waveform in signaldependent interference and waveform design technique for target detection and estimation in signal-dependent interference. Section 5 shows some simulation results. The whole paper is summarized in Section 6.

### 2. Signal Model

The block diagram in Figure 1 represents the signal model of a target ensemble in ground clutter being considered. Let x(t) be a finite-energy waveform with duration T. Let g(t)be a zero-mean extended target with energy spectral variance  $\sigma_G^2(f)$ . Let  $T_g$  be the time duration where most of the target impulse's energy resides. It is necessary to have  $T_g > T$  to capture the target impulse response's energy. The clutter c(t)is a zero-mean complex Gaussian random process with power spectral density (PSD)  $\sigma_C^2(f)$ , and n(t) is the zero-mean receiver noise process with one-sided PSD  $P_n(f)$ . In addition, n(t) is assumed to be statistically independent of the transmitted waveform x(t), the target impulse response g(t), and the clutter c(t).

The waveform received at the receiver is filtered by the ideal lowpass filter B(f), passing only frequencies in the band  $\omega$ . This is just a statement of the fact that we assume that the transmitted signal has no significant energy outside the frequency interval w = [-W, W]. Since z(t) and d(t) are the response of a linear time-invariant system to the transmitted signal, they do not have significant energy outside the frequency interval w = [-W, W]. Hence we will not consider frequencies outside this interval.

Let y(t) be the received signal given by

$$y(t) = z(t) + d(t) + n(t).$$
(1)

z(t) and d(t) are defined by

$$z(t) = x(t) * g(t),$$
 (2)

$$d(t) = x(t) * c(t),$$

where \* denotes the convolution operator.

### 3. Constant Envelope Signals with Given Fourier Transform Magnitude

Pillai et al. give us a technique of generating a constant modulus signal with the given Fourier transform magnitude in [11]. It is summarized as follows.

Let  $C_M$  denote the set of functions  $\{g(t)\}$  that have the prescribed Fourier transform magnitude  $M(\omega)$  over a prescribed frequency set  $\Omega$ . The operator  $P_M$  will assign every



FIGURE 1: Signal model of a target ensemble in signal-dependent interference.

arbitrary function x(t) a "nearest neighbor"  $P_M x(t)$  that belongs to  $C_M$  such that there exists no other element  $g \in C_M$  for which  $||x - g|| < ||x - P_M x||$  is satisfied.

Given an arbitrary function x(t), its corresponding Fourier transform is  $X(\omega) = |X(\omega)|e^{j\Omega(\omega)}$  and the magnitude projection of x(t) is defined as

$$P_M x(t) \longleftrightarrow \begin{cases} M(\omega) e^{j\Omega(\omega)}, & \omega \in \Omega\\ X(\omega), & \omega \in \Omega'. \end{cases}$$
(3)

For a constant envelope signal x(t), it can be expressed as

$$x(t) = Ae^{j\theta(t)},\tag{4}$$

where *A* is a suitable positive constant that can be used to maintain a prescribed energy level for x(t).

Interestingly, constant envelope signals also share properties similar to the Fourier transform magnitude situation. Notice that if  $C_A$  denotes the set of functions  $\{g(t)\}$  which have constant envelope level A, the operator  $P_A$  will assign every arbitrary function x(t) a nearest neighbor  $P_A x(t)$  that belongs to  $C_A$  such that no other element  $g \in C_A$  satisfies  $||x - g|| < ||x - P_A x||$ .

Given an arbitrary signal  $x(t) = a(t)e^{j\theta(t)}$ , the projection procedure is

$$P_{A}x(t) = \begin{cases} Ae^{j\theta(t)}, & t \in T\\ x(t), & \text{otherwise.} \end{cases}$$
(5)

The magnitude and amplitude projection are combined according to

$$x_{k+1} = P_A P_M x_k, \tag{6}$$

where  $x_k$  is the *k*th iterative function. After a number of magnitude and amplitude projections, the function  $x_k$  satisfies the constant modulus property exactly while approximately maintaining the prescribed Fourier transform magnitude.

### 4. Waveform Design Based on Constant Modulus Constraint

We note that x(t) is a deterministic waveform. It is explicitly denoted in I(y(t); g(t) | x(t)) because the mutual information is a function of x(t), and we are interested in finding those functions x(t) that maximize I(y(t); g(t) | x(t)) under constrains on their energy and bandwidth.

Bell proposed the derivation of the mutual information in the channel-noise-only case and derived the informationbased waveform solution. Here we provide the derivation of the mutual information in the presence of signal-dependent clutter.

Here we have a channel (as shown in Figure 2) with input Z (a zero-mean Gaussian random variable with variance  $\sigma_Z^2$ ), clutter D (a zero-mean Gaussian random variable with variance  $\sigma_D^2$ ), and additive zero-mean Gaussian noise N with variance  $\sigma_N^2$ . The mutual information I(Y; Z) between Y and Z is

$$I(Y;Z) = H(Y) - H(Y \mid Z).$$
 (7)

The differential entropies H(Y) and  $H(Y \mid Z)$  are

$$H(Y) = \frac{1}{2} \ln 2\pi \sigma_Y^2 = \frac{1}{2} \ln 2\pi \left( \sigma_Z^2 + \sigma_N^2 + \sigma_D^2 \right),$$
  

$$H(Y \mid Z) = \frac{1}{2} \ln 2\pi \left( \sigma_N^2 + \sigma_D^2 \right).$$
(8)

Thus the mutual information is given by the expression

$$I(Y;Z) = H(Y) - H(Y \mid Z) = \frac{1}{2} \ln\left(1 + \frac{\sigma_Z^2}{\sigma_N^2 + \sigma_D^2}\right).$$
 (9)

Consider again the signal model of Figure 1. Assume that  $\hat{z}_k(t)$ ,  $\hat{y}_k(t)$ ,  $\hat{d}_k(t)$ , and  $\hat{n}_k(t)$  are the sample signal in the frequency band  $F_k = [f_k, f_k + \Delta f]$  and the sampling rate is  $2\Delta f$ . The samples  $\hat{z}_k(t)$  are independent, identically distributed random variables with zero mean and variance  $\sigma_Z^2$ . Note that the total energy of  $\hat{z}_k(t)$  is

$$\varepsilon_Z(F_k) = 2\Delta f \left| X(f_k) \right|^2 \sigma_G^2(f_k).$$
<sup>(10)</sup>

The factor 2 in the previous formula is due to the fact that  $X(f_k)$  is the two-sided spectrum of x(t) and that we are carrying out our calculations using only positive frequencies. In the time interval  $T_y$ , the total samples statistically independent are  $2\Delta f T_y$ . So the variance of each sample is

$$\sigma_{Z}^{2} = \frac{\varepsilon_{Z}(F_{k})}{2\Delta f T_{y}} = \frac{2\Delta f |X(f_{k})|^{2} \sigma_{G}^{2}(f_{k})}{2\Delta f T_{y}} = \frac{|X(f_{k})|^{2} \sigma_{G}^{2}(f_{k})}{T_{y}}.$$
(11)

The clutter process has the total energy on the interval  $T_y$  given by

$$\varepsilon_D(F_k) = 2\Delta f \left| X(f_k) \right|^2 \sigma_c^2(f_k) T_y.$$
(12)

The variance of each sample is

$$\sigma_D^2 = \frac{\varepsilon_D(F_k)}{2\Delta f T_y} = \frac{2\Delta f |X(f_k)|^2 \sigma_C^2(f_k) T_y}{2\Delta f T_y}$$
(13)  
=  $|X(f_k)|^2 \sigma_C^2(f_k).$ 



FIGURE 2: Channel model in the presence of clutter and additive Gaussian noise.

The noise process has the total energy on the interval  $T_y$  given by

$$\varepsilon_N(F_k) = \Delta f P_n(f) T_{\gamma}. \tag{14}$$

Hence, the variance  $\sigma_N^2$  of each sample is

$$\sigma_N^2 = \frac{\varepsilon_N(F_k)}{2\Delta f T_y} = \frac{\Delta f P_n(f) T_y}{2\Delta f T_y} = \frac{P_n(f)}{2}.$$
 (15)

The mutual information between each sample  $Z_m$  of  $\hat{z}_k(t)$ and the corresponding sample  $Y_m$  of  $\hat{y}_k(t)$  is

$$I(Y_m; Z_m) = \frac{1}{2} \ln \left[ 1 + \frac{2|X(f_k)|^2 \sigma_G^2(f_k)}{T_y \left\{ P_n(f_k) + 2|X(f_k)|^2 \sigma_C^2(f_k) \right\}} \right].$$
(16)

Now there are  $2\Delta f T_y$  statistically independent sample values for both  $\hat{z}_k(t)$  and  $\hat{y}_k(t)$  in the observation interval  $T_y$ . Thus,

$$I\left(\hat{y}_{k}(t); \hat{z}_{k}(t) \mid x(t)\right)$$
  
=  $2\Delta f T_{y} I\left(Y_{m}; Z_{m}\right)$   
=  $\Delta f T_{y} \ln\left[1 + \frac{2|X(f_{k})|^{2}\sigma_{G}^{2}(f_{k})}{T_{y}\left\{P_{n}(f_{k}) + 2|X(f_{k})|^{2}\sigma_{C}^{2}(f_{k})\right\}}\right].$  (17)

If we now consider the frequency interval  $\omega = [0, W]$ , partition it into a large number of disjoint intervals of bandwidth  $\Delta f$ ; then let the number of intervals increase as  $\Delta f \rightarrow$ 0, in the limit we obtain an integral for the mutual information I(y(t); z(t) | x(t)), where we assume the x(t), y(t), and z(t) are confined to the frequency interval  $\omega$ . Hence the mutual information I(y(t); z(t) | x(t)) is

$$I(y(t); z(t) | x(t)) = T_{y} \int_{W} \ln\left(1 + \frac{2|X(f)|^{2}\sigma_{G}^{2}(f)}{T_{y} \left\{P_{n}(f) + 2|X(f)|^{2}\sigma_{C}^{2}(f)\right\}}\right) df,$$
(18)

as

$$I(y(t); g(t) | x(t)) = I(y(t); z(t) | x(t)).$$
(19)

Thus the mutual information between the random target impulse response and the received radar waveform is

$$I(y(t); g(t) | x(t)) = T_{y} \int_{W} \ln \left( 1 + \frac{2|X(f)|^{2} \sigma_{G}^{2}(f)}{T_{y} \left\{ P_{n}(f) + 2|X(f)|^{2} \sigma_{C}^{2}(f) \right\}} \right) df.$$
(20)

For the MI waveform derivation, we treat the receiver filter as an ideal lowpass filter with approximate time duration  $T_B \leq T$  and  $T_B \leq T_g$ . Therefore  $T_B$  can be effectively ignored, and the receive filter simply becomes an explicit statement that the radar system is band limited. Therefore,  $T_v$  is

$$T_{y} = T + T_{g}.$$
(21)

The mutual information between the random target impulse response and the received radar waveform is shown in formula (20). The output SINR is defined to be the ratio of the average power of the signal component to the average power of the noise and interference component [12]. Thus, SINR is expressed as

SINR = 
$$\int_{W} \frac{2|X(f)|^2 \sigma_G^2(f)}{P_n(f) + 2|X(f)|^2 \sigma_C^2(f)} df.$$
 (22)

We can assume a lower bound  $\mathrm{SINR}_0$  on the SINR for the target

$$\int_{W} \frac{2|X(f)|^{2} \sigma_{G}^{2}(f)}{P_{n}(f) + 2|X(f)|^{2} \sigma_{C}^{2}(f)} df \ge \text{SINR}_{0}.$$
 (23)

The energy constraint in the band  $\omega = [0, W]$  is expressed as

$$\int_{W} \left| X(f) \right|^2 df \le E_x.$$
(24)

With these constraints in mind, we can now formulate the arbitrary waveform design problem as the following constrained optimization problem:

$$\max \quad T_{y} \int_{W} \ln \left( 1 + \frac{2|X(f)|^{2} \sigma_{G}^{2}(f)}{T_{y} \left\{ 2|X(f)|^{2} \sigma_{C}^{2}(f) + P_{n}(f) \right\}} \right) df$$
  
s.t. 
$$\int_{W} \frac{2|X(f)|^{2} \sigma_{G}^{2}(f)}{P_{n}(f) + 2|X(f)|^{2} \sigma_{C}^{2}(f)} df \ge \text{SINR}_{0}$$
$$\int_{W} |X(f)|^{2} df - E_{x} \le 0.$$
(25)

The previous constrained problem can be formulated as a convex optimization problem by introducing the autocorrelation sequence of the transmitted signal. Then an interiorpoint method can be used to carry out the optimization task.



FIGURE 3: Closed-loop radar system.

Although the solution to this formulation is local optimal, this process is complicated. We need to solve the nonlinear constrained maximization problem.

After applying the technique of generating a constant modulus signal with the given Fourier transform magnitude to the above obtained waveform, we can get a waveform under multiple constraints.

Figure 3 represents the closed-loop radar system in signal-dependent interference proposed for target detection and estimation. In this figure, the CR signal processing involved is best described by a block labeled "COGNITIVE RADAR". CR is an intelligent system. Through sensing the environment, CR transmits the waveform suited to the working conditions. The radar returns and environment factors help to construct the new waveform that achieves a trade-off between the mutual information and the SINR for the target, that is, an optimal trade-off between the detection and estimation criteria. Then it reconstructs a signal with constant envelope property in the time domain according to its Fourier transform magnitude. The signal satisfies the constant modulus property exactly while approximately maintaining the prescribed Fourier transform magnitude. Then the waveform is transmitted to the environment. It forms a feed-back loop, and the cycle goes on and on.

### 5. Simulation

We consider an arbitrary target spectrum and clutter spectrum shown in Figure 4. The total energy is 1. The noise variance is 0.1. The lower bound  $SINR_0$  is -8 dB. The number of sample points is 128. Sampling frequency is 2. Modulus value is 0.25.

Figure 5 is energy spectrum of unconstrained waveform. It shows that the optimized radar waveform only selects the dominant frequency components of the target spectrum. However, it does not distribute energy among different modes of the target. Investigating the reason, there are approximately two: one is the spectrum amplitudes scale in order to compensate for the clutter spectrum, and the other is to balance the detection performance. Hence it provides an optimal trade-off between the detection and estimation criteria.

Figure 6 is energy spectrum of constant modulus constrained waveform. It shows that the constant modulus constraint spreads the waveform energy into additional frequency bands, but the four peak amplitudes are maintained. The energy spectrum of constant modulus constrained waveform in Figure 6 is similar to the energy spectrum of unconstrained waveform in Figure 5. Thus it guarantees the performance of the nonconstant modulus optimized waveform.



FIGURE 4: Target spectrum and clutter spectrum.



FIGURE 5: Energy spectrum of unconstrained waveform.



FIGURE 6: Energy spectrum of constant modulus constrained waveform.



FIGURE 7: Complex constellation of constant modulus constrained waveform.

Figure 7 shows the time domain representation of the signal in the complex domain with real and imaginary parts of each instant plotted as *x*-axis and *y*-axis. The figure shows that after applying the constant modulus constraint, the temporal waveform has constant amplitude. Thus the transmitted waveform has no longer high peak amplitude in time domain and can effectively through DAC and PA of transmitter.

### 6. Conclusions

In this paper, we investigate the mutual information between the target impulse response and received radar waveform in signal-dependent interference and channel noise. Then we discuss the problem of radar waveform design under multiple constraints. Here we consider a situation when the signaldependent interference is not negligible. An optimal tradeoff between the detection and estimation criteria is provided. After applying the technique of generating a constant modulus signal with the given Fourier transform magnitude to the optimal waveform, a waveform that has constant modulus property is obtained. Simulation results have a significant meaning in the waveform design in cognitive radar. They show that the energy spectrum of constant modulus constrained waveform is similar to the energy spectrum of unconstrained waveform. Hence the performance of the nonconstant modulus optimized waveform is guaranteed. The waveform can also be applied to a CR performing target identification.

### Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (no. N110323005), the Natural Science Foundation of Hebei Province (no. F2013501075), and the Doctoral Scientific Research Foundation of Liaoning Province (no. 20131030).

### References

- S. Haykin, "Cognitive radar: a way of the future," *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 30–40, 2006.
- [2] S. Haykin, "Cognition is the key to the next generation of radar systems," in *Proceedings of the Digital Signal Processing* Workshop and 5th IEEE Signal Processing Education Workshop (DSP/SPE '09), pp. 463–467, IEEE Press, January 2009.
- [3] S. U. Pillai, H. S. Oh, D. C. Youla, and J. R. Guerci, "Optimum transmit-receiver design in the presence of signal-dependent interference and channel noise," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 577–584, 2000.
- [4] R. A. Romero, J. Bae, and N. A. Goodman, "Theory and application of SNR and mutual information matched illumination waveforms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 912–927, 2011.
- [5] M. R. Bell, "Information theory and radar waveform design," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1578–1597, 1993.
- [6] A. Leshem, O. Naparstek, and A. Nehorai, "Information theoretic adaptive radar waveform design for multiple extended targets," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 42–55, 2007.
- [7] Y. Nijsure, Y. Chen, P. Rapajic, C. Yuen, Y. H. Chew, and T. F. Qin, "Information-theoretic algorithm for waveform optimization within ultra wideband cognitive radar network," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB '10)*, pp. 1–4, IEEE Press, September 2010.
- [8] S. Haykin, Y. B. Xue, and T. N. Davidson, "Optimal waveform design for cognitive radar," in *Proceedings of the 42nd Asilomar Conference on Signals*, pp. 3–7, Systems and Computers, Pacific Grove, Calif, USA, October 2008.
- [9] H. B. Jun and N. A. Goodman, "Adaptive waveforms for target class discrimination," in *Proceedings of the International Waveform Diversity and Design Conference*, pp. 395–399, June 2007.
- [10] L. K. Patton and B. D. Rigling, "Modulus constraints in adaptive radar waveform design," in *Proceedings of the IEEE Radar Conference (RADAR '08)*, pp. 1–6, IEEE Press, Rome, Italy, May 2008.
- [11] S. U. Pillai, Y. L. Ke, and H. Beyer, "Reconstruction of constant envelope signals with given Fourier transform magnitude," in *Proceedings of the IEEE Radar Conference (RADAR '09)*, pp. 1– 4, IEEE Press, Pasadena, Calif, USA, May 2009.
- [12] R. A. Romero and N. A. Goodman, "Waveform design in signaldependent interference and application to target recognition with multiple transmissions," *IET Radar, Sonar and Navigation*, vol. 3, no. 4, pp. 328–340, 2009.

# Research Article **Optimal Waveform Selection for Robust Target Tracking**

### Fengming Xin, Jinkuan Wang, Qiang Zhao, and Yuhuan Wang

School of Information Science and Engineering, Northeastern University, Shenyang 110004, China

Correspondence should be addressed to Jinkuan Wang; wjk@mail.neuq.edu.cn

Received 15 May 2013; Accepted 28 June 2013

Academic Editor: Bin Wang

Copyright © 2013 Fengming Xin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a new optimal waveform selection algorithm for intelligent target tracking. In radar systems, optimal waveform is inspired by the improvements in performance. When the target is intelligent and tries to escape from detection, it will maximize the estimation error to degrade the target tracking performance. So the conventional tracking algorithms are not suitable for this situation. In this paper, we assume a one-dimension target model which will try to escape the radar detection to degrade the tracking performance. A new optimal waveform selection algorithm is proposed based on game theory for robust tracking. The robust received filter is first reviewed according to zero-sum game with the derivation of estimated state error covariance. The parameters for transmitted waveform that need to be optimized are found to be related to the robust filter. The optimal parameters for transmitted waveform are finally found by the minimization of the trace of the estimated state error covariance. Simulation results show the effectiveness of this new proposed algorithm for optimal waveform selection for intelligent target tracking.

### 1. Introduction

Since traditional radar/sonar systems lack adaptivity to the different targets, interference, and clutter without utilizing prior measurements or knowledge, they could not adaptively adjust transmitted waveforms to the variant environment. So, the modern radar/sonar systems need more intelligent ability in order to improve the radar performance. Cognitive radar is proposed as a new generation radar system by Haykin [1, 2], which can adaptively and intelligently interrogate a propagation channel using all available knowledge. The most important conclusion of cognitive radar system is that it must be able to adaptively generate and transmit the optimal waveforms to improve the accuracy of the radar system.

There are two strategies of generating optimal waveforms, that is, selection and design. However, it is not clear which one is better. Many researchers focused on the optimal waveform technology for different tasks, for example, target detection, estimation, and target tracking [3–17]. The general method is to find signal/filter pairs to maximize the signal-to-clutter plus interference ratio (SCIR) for detecting the target. Pillai et al. developed an eigensolution for optimal signal/filter pairs for target detection when the target and clutter can be seen as linear time invariant random processes [3, 4].

Then they extended this approach to optimize the waveform for target identification. The waveform optimization for target identification is addressed by relating SCIR to the Mahalanobis distance [5, 6]. Information theoretic approach is also an important tool for the waveform optimization. Bell proposed to maximize the mutual information (MI) between the received signal and target impulse response to optimize the waveform [7]. In [8], the authors introduced the relative entropy to optimal waveform for target identification based upon the synthesis of a sequence of probing signals to maximize classification performance, which can extract as much information as possible from the observations. Kay derived the optimal NP detector firstly, which shows that the NP detection performance does not immediately lead to an obvious signal design criterion so that a divergence criterion is proposed for signal design, also based on the relative entropy in signal input multiple output radar scenario [9]. Goodman et al. adopted sequential hypothesis testing combination with mutual information and maximizing the signal-to-noise ratio (SNR) that decides when hard decision may be made with adequate confidence to design the waveform [10]. By comparing the performance of two different waveform design techniques based on information theory [7] and eigensolution [5], Romero et al. also found the relationship between the MI and maximizing the SNR in the context of waveform design for stochastic target [11]. In [12], the authors extended Goodman's method, which considered the target detection before the recognition procedure.

There are two main approaches of designing the optimal waveform for target tracking, that is, the control theoretic and information theoretic approaches. The first one is treated as a control problem, since the parameters of the transmitted waveform as an input vector, which is selected or designed to affect the next observation and the tracker, update in a feedback loop. In [13], the authors created the cost function that includes the parameters of transmitted waveform for the next step. The second one also made use of the mutual information (MI) between the target and the observations [14, 15]. In [14], the authors designed the libraries where the waveform was selected through maximizing the mutual information (MI) between the target model and observations. Then they extended this method to interacting multiple model trackers for different dynamic models [15]. In [13], Kershaw and Evans used the control theoretic approach to optimize the waveform for one-dimensional target tracking in a feedback loop system. They derived the Cramer-Rao lower bound (CRLB) for estimating error variance from the curvature of the peak of the ambiguity function (AF). then the measurement noise covariance matrix that is related to the parameters of the transmitted waveform can be evaluated from the CRLB in high SNR condition. The minimization of the tracking mean square error and the validation gate volume are performed to select the next transmitted waveform. Kershaw and Evans extended their work in clutter and imperfect detection situation [16]. This method was also introduced to the wideband environment for multiple targets tracking in clutter condition [17].

In this paper, we adopt the control theoretic approach to find the optimal waveform for one dimension target tracking [13]. In this paper, we assume a one-dimension target model which will try to escape the radar detection to degrade the tracking performance. A new optimal waveform selection algorithm is proposed based on game theory for robust tracking [18]. The robust received filter is first reviewed according to zero-sum game with the derivation of estimated state error covariance. The parameters for transmitted waveform that need to be optimized are found to be related to the robust filter. The optimal parameters for transmitted waveform are finally found by the minimization of the trace of the estimated state error covariance. Simulation results show the effectiveness of this new proposed algorithm for optimal waveform selection for intelligent target tracking.

This paper is organized in the following manner. Section 2 reviews the control approach in [13] and presents the problem for target tracking. Section 3 describes the optimal waveform selection for robust target tracking. Section 4 shows the simulation results. The conclusion is summarized in Section 5.

### 2. Problem Formulation

We begin with a brief overview of the control approach for one-dimension target tracking in [13]. In radar/sonar system, the transmitted signal can be written as

$$s_T(t) = \sqrt{2} \operatorname{Re}\left\{\sqrt{E_T}\tilde{s}(t) e^{j\omega_c t}\right\},\qquad(1)$$

where  $\tilde{s}(t)$  is the complex envelope,  $\omega_c$  is the carrier frequency, and  $E_T$  is the energy of the transmitted signal. When the target exists, the received waveform envelope is

$$\overline{r}(t) = \sqrt{E_R} e^{j\phi} \widetilde{s}(t - \tau_0) e^{j\nu_o t} + \widetilde{n}(t), \qquad (2)$$

where  $E_R$  is the energy of the received signal.  $\tilde{n}(t)$  is zeromean complex white Gaussian noise with real spectral density  $N_0/2$ .  $\tau_0$  and  $v_0$  denote the target time delay and Doppler shift, respectively.

The ambiguity function corresponding to the received waveform in frequency domain is written as

$$A(\tau,\nu) = \int_{-\infty}^{\infty} S\left(\omega - \frac{\nu}{2}\right) S^*\left(\omega + \frac{\nu}{2}\right) e^{-j\omega\tau} d\omega/2\pi.$$
 (3)

The receiver parameter vector is  $\boldsymbol{\alpha} = [\tau, \nu]^T$ . The target state model as discrete time is defined by

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k. \tag{4}$$

The measurement vector equation is given as

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k,\tag{5}$$

where  $\mathbf{y}_k = [\mathbf{r}, \dot{\mathbf{r}}]$  denotes the measurement vector. r is the range and  $\dot{r}$  is the target velocity.  $\mathbf{x}_k$  is the target state vector at time k. **F**, **G**, and **H** are given matrices for onedimensional target tracking.  $\mathbf{w}_k$  and  $\mathbf{n}_k$  are the zero-mean white Gaussian noise vectors with covariance matrices  $\mathbf{Q}_k$  and  $\mathbf{N}(\boldsymbol{\theta}_k)$ , respectively. The vector  $\boldsymbol{\theta}_k$  characterizes the waveform parameters at time k.

According to Lemma 3.1 in [13], build the relationship between the receiver estimation parameter vector  $\boldsymbol{\alpha}$  and measurement vector  $\mathbf{y}$  through a linear transformation  $\mathbf{T}$ , that is,  $\mathbf{y} = \mathbf{T}\boldsymbol{\alpha}$ . And the measurement noise covariance matrix is dependent on waveform parameter  $\boldsymbol{\theta}$  as follows:

$$\mathbf{N}\left(\boldsymbol{\theta}\right) = \frac{1}{\eta} \mathbf{T} \mathbf{J}^{-1}\left(\boldsymbol{\theta}\right) \mathbf{T},\tag{6}$$

where  $\mathbf{T} = \text{diag}(c/2, c/2\omega_c)$ , **J** is the Fisher information and  $\text{Cov}(\boldsymbol{\alpha}) = \mathbf{J}^{-1}(\boldsymbol{\theta})$ .

After finding the relationship between the measurement noise covariance matrix and the waveform parameter, the Kalman filter equations are dependent on  $\theta$  as follows:

$$\mathbf{S}_{k}(\boldsymbol{\theta}_{k}) = \mathbf{H}\mathbf{P}_{k/k-1}\mathbf{H}^{T} + \mathbf{N}(\boldsymbol{\theta}_{k}),$$

$$\mathbf{K}_{k}(\boldsymbol{\theta}_{k}) = \mathbf{P}_{k/k-1}\mathbf{H}^{T}\mathbf{S}_{k}^{-1}(\boldsymbol{\theta}_{k}),$$

$$\widehat{\mathbf{x}}_{k/k}(\boldsymbol{\theta}_{k}) = \widehat{\mathbf{x}}_{k/k-1} - \mathbf{K}_{k}(\boldsymbol{\theta}_{k})(\mathbf{y}_{k} - \mathbf{H}\widehat{\mathbf{x}}_{k/k-1}),$$

$$\mathbf{P}_{k/k}(\boldsymbol{\theta}_{k}) = \mathbf{P}_{k/k-1}(\boldsymbol{\theta}_{k}) - \mathbf{K}_{k}(\boldsymbol{\theta}_{k})\mathbf{S}_{k}(\boldsymbol{\theta}_{k})\mathbf{K}_{k}^{T}(\boldsymbol{\theta}_{k}),$$

$$\widehat{\mathbf{x}}_{k+1/k}(\boldsymbol{\theta}_{k}) = \mathbf{F}\widehat{\mathbf{x}}_{k/k}(\boldsymbol{\theta}_{k}),$$

$$\mathbf{P}_{k+1/k}(\boldsymbol{\theta}_{k}) = \mathbf{F}\mathbf{P}_{k/k}(\boldsymbol{\theta}_{k})\mathbf{F}^{T} + \mathbf{G}\mathbf{Q}_{k}\mathbf{G}^{T}.$$
(7)

In order to improve the tracking performance, minimizing the trace of the mean square tracking error as cost function is used to select the next transmitted waveform. That is,

$$\boldsymbol{\theta}_{k+1}^{*} = \arg\min_{\boldsymbol{\theta}_{k+1} \in \Theta} \operatorname{Tr} \left\{ \mathbf{P}_{k+1/k+1} \left( \boldsymbol{\theta}_{k+1} \right) \right\}.$$
(8)

In addition, minimization of the validation gate volume as another cost function is to select next transmitted waveform, which will reduce the number of false alarms in high SNR or clutter environment. So the next transmitted waveform is determined by

$$\boldsymbol{\theta}_{k+1}^* = \arg\min_{\boldsymbol{\theta}_{k+1} \in \Theta} \det \left\{ \mathbf{S}_{k+1} \left( \boldsymbol{\theta}_{k+1} \right) \right\}.$$
(9)

When the target is intelligent enough to maximize the estimation error, it could deliberately degrade the tracking performance and even break the tracking task down. In this case, the target state model has a fictitious adversary disturbance that includes some unknown noise, which could be "smart" enough to maximize the estimation state error and decrease the target tracking performance [18]. Thus the Kalman filter and its relative optimal waveform method mentioned before are not suitable for this case. Thus, we should consider the robust tracking problem, a minimax filter based on zero-sum game is needed for target tracking, and a new method for optimal waveform will be presented in Section 3.

### 3. Minimax Filter and Waveform Selection

*3.1. Minimax Filter.* In order to guarantee the target tracking performance for "smart" target, the minimax filter is needed. Like [18, 19], the discrete linear time-invariant system in adversary disturbance which existed is expressed by

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k + \mathbf{d}_k,\tag{10}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k,\tag{11}$$

where

$$\mathbf{d}_{k} = \mathbf{L}_{k} \left( \mathbf{H} \left( \mathbf{x}_{k} - \widehat{\mathbf{x}}_{k/k-1} \right) + \mathbf{v}_{k} \right).$$
(12)

Equation (12) is adversary disturbance signal which could increase the estimated error. **L** is a gain to be determined;  $\mathbf{v}_k$  is Gaussian noise vector with zero mean and covariance matrix **R**. The other parameters, **x**, **y**, **H**, **G**, **w**, and **n**, in (10) and (11), are the same as in (4) and (5).

Based on zero-sum game, the predicted state is

$$\widehat{\mathbf{x}}_{k+1/k} = \mathbf{F}\widehat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k \left( \mathbf{y}_k - \mathbf{H}\widehat{\mathbf{x}}_{k/k-1} \right), \tag{13}$$

where  $\hat{\mathbf{x}}_{k+1/k}$  is the predicted state, **K** is the minimax filter gain, and the prediction state error is defined by

$$\mathbf{e}_{k/k-1} = \mathbf{x}_k - \widehat{\mathbf{x}}_{k/k-1}.$$
 (14)

Substituting (10) and (13) into (14), we have

$$\mathbf{e}_{k+1/k} = \mathbf{F} \left( \mathbf{x}_k - \widehat{\mathbf{x}}_{k/k-1} \right) - \mathbf{K}_k \left( \mathbf{y}_k - \mathbf{H} \widehat{\mathbf{x}}_{k/k-1} \right) + \mathbf{G} \mathbf{w}_k + \mathbf{d}_k.$$
(15)

$$\mathbf{e}_{k+1/k} = (\mathbf{F} - \mathbf{K}\mathbf{H} + \mathbf{L}\mathbf{H}) \,\mathbf{e}_{k/k-1} + \mathbf{G}\mathbf{w}_k + \mathbf{L}\mathbf{v}_k - \mathbf{K}\mathbf{n}_k.$$
 (16)

From (15), it can been seen that the adversary part,  $\mathbf{d}_k$ , can increase the estimation error. To prevent this, the estimation error in (16) can be decomposed as follows:

$$\mathbf{e}_{k+1/k} = \mathbf{e}_{k+1/k}^{K} + \mathbf{e}_{k+1/k}^{L},$$
 (17)

where

$$\mathbf{e}_{k+1/k}^{K} = \left(\mathbf{F} - \mathbf{K}_{k}\mathbf{H} + \mathbf{L}_{k}\mathbf{H}\right)\mathbf{e}_{k/k-1}^{K} + \mathbf{G}\mathbf{w}_{k} - \mathbf{K}_{k}\mathbf{n}_{k},$$
$$\mathbf{e}_{0}^{K} = \mathbf{x}_{0},$$
$$\mathbf{e}_{k+1/k}^{L} = \left(\mathbf{F} - \mathbf{K}_{k}\mathbf{H} + \mathbf{L}_{k}\mathbf{H}\right)\mathbf{e}_{k/k-1}^{L} + \mathbf{L}_{k}\mathbf{v}_{k},$$
$$\mathbf{e}_{0}^{L} = \mathbf{0}.$$
(18)

Motivated by [18, 19] the cost function is defined by

$$J(\mathbf{K}, \mathbf{L}) = \operatorname{trace}\left(\sum_{k=0}^{T} E\left[\left\|\mathbf{e}_{k+1/k}^{K}\right\|^{2} - \left\|\mathbf{e}_{k+1/k}^{L}\right\|^{2}\right]\right).$$
(19)

The minimax filter designed based on zero-sum game is to find the optimized filter gain **K** and the robust filter gain **L**. The gain **K** should be optimized to minimize the *J* so that the tracking performance is better, since the prediction error  $\mathbf{e}_{k+1/k}^{K}$  is relative to noises of  $\mathbf{w}_{k}$  and  $\mathbf{n}_{k}$ . The gain **L** should be optimized to maximize the *J*, since the  $\mathbf{e}_{k+1/k}^{L}$  is relative to the noise of  $\mathbf{v}_{k}$ , which makes the worst possible disturbance. Let  $\mathbf{K}^{*}$  and  $\mathbf{L}^{*}$  denote the optimized gains, which satisfies a saddle-point equilibrium, that is,

$$J\left(\mathbf{K}^{*},\mathbf{L}\right) \leq J\left(\mathbf{K}^{*},\mathbf{L}^{*}\right) \leq J\left(\mathbf{K},\mathbf{L}^{*}\right).$$
(20)

To solve (20), the cost function (19) needs to be written in a more convenient form. Define  $Z_k$  as follows:

$$\mathbf{Z}_{k} = \mathbf{F} - \mathbf{K}_{k}\mathbf{H} + \mathbf{L}_{k}\mathbf{H}.$$
 (21)

Let  $\mathbf{P}_{k+1/k}^{K} = E[(\mathbf{e}_{k+1/k}^{K})(\mathbf{e}_{k+1/k}^{K})^{T}]$  and  $\mathbf{P}_{k+1/k}^{L} = E[(\mathbf{e}_{k+1/k}^{L})(\mathbf{e}_{k+1/k}^{L})^{T}]$ ; we have

$$\mathbf{P}_{k+1/k}^{K} = \mathbf{Z}_{k} \mathbf{P}_{k/k-1}^{K} \mathbf{Z}_{k}^{T} + \mathbf{G} \mathbf{Q}_{k} \mathbf{G}^{T} + \mathbf{K}_{k} \mathbf{N} \mathbf{K}_{k}^{T},$$

$$\mathbf{P}_{k+1/k}^{L} = \mathbf{Z}_{k} \mathbf{P}_{k/k-1}^{L} \mathbf{Z}_{k}^{T} + \mathbf{L}_{k} \mathbf{R} \mathbf{L}_{k}^{T}.$$
(22)

The cost function (19) can be rewritten by

$$J(\mathbf{K}, \mathbf{L}) = \operatorname{trace}\left(\sum_{k=0}^{T} \mathbf{P}_{k+1/k}\right), \qquad (23)$$

where  $\mathbf{P}_{k+1/k} = \mathbf{P}_{k+1/k}^K - \mathbf{P}_{k+1/k}^L = \mathbf{Z}_k \mathbf{P}_{k/k-1} \mathbf{Z}_k^{\%} + \mathbf{G} \mathbf{Q}_k \mathbf{G}^T + \mathbf{K}_k \mathbf{N} \mathbf{K}_k^T - \mathbf{L}_k \mathbf{R} \mathbf{L}_k^T.$ Let

$$\mathbf{U}_{k} = \mathbf{G}\mathbf{Q}_{k}\mathbf{G}^{T} + \mathbf{K}_{k}\mathbf{N}\mathbf{K}_{k}^{T} - \mathbf{L}_{k}\mathbf{R}\mathbf{L}_{k}^{T}.$$
 (24)

Then  $\mathbf{P}_{k+1/k} = \mathbf{Z}_k \mathbf{P}_{k/k-1} \mathbf{Z}_k^T + \mathbf{U}_k$ . According to Theorem 1 in [18], the game equilibrium is derived by

$$\mathbf{K}_{k}^{*} = \mathbf{F} \boldsymbol{\Sigma}_{k} \mathbf{H}^{T} \mathbf{N}^{-1},$$

$$\mathbf{L}_{k}^{*} = \mathbf{F} \boldsymbol{\Sigma}_{k} \mathbf{H}^{T} \mathbf{R}^{-1},$$

$$\boldsymbol{\Sigma}_{k}^{-1} = \mathbf{P}_{k/k-1}^{-1} + \mathbf{H}^{T} \left( \mathbf{N}^{-1} - \mathbf{R}^{-1} \right) \mathbf{H}.$$
(25)

After obtaining the game equilibrium  $(\mathbf{K}_k^*, \mathbf{L}_k^*)$ , substitute  $\mathbf{K}_k^*$  and  $\mathbf{L}_k^*$  into (21) and (24), respectively. the covariance matrix  $\mathbf{P}_{k+1/k}$  can be expressed by

$$\mathbf{P}_{k+1/k} = \mathbf{F} \boldsymbol{\Sigma}_k \mathbf{F}^T + \mathbf{G} \mathbf{Q} \mathbf{G}^T.$$
(26)

Finally, the minimax filter based on zero-sum game equations is

$$\widehat{\mathbf{x}}_{k+1/k} = \mathbf{F}\widehat{\mathbf{x}}_{k/k-1} + \mathbf{K}_{k}^{*} \left(\mathbf{y}_{k} - \mathbf{H}\widehat{\mathbf{x}}_{k/k-1}\right),$$

$$\mathbf{P}_{k+1/k} = \mathbf{F}\Sigma_{k}\mathbf{F}^{T} + \mathbf{G}\mathbf{Q}\mathbf{G}^{T},$$

$$\Sigma_{k}^{-1} = \mathbf{P}_{k/k-1}^{-1} + \mathbf{H}^{T} \left(\mathbf{N}^{-1} - \mathbf{R}^{-1}\right)\mathbf{H},$$

$$\mathbf{K}_{k}^{*} = \mathbf{F}\Sigma_{k}\mathbf{H}^{T}\mathbf{N}^{-1},$$

$$\mathbf{L}_{k}^{*} = \mathbf{F}\Sigma_{k}\mathbf{H}^{T}\mathbf{R}^{-1}.$$
(27)

The minimax filter based on the game theory is suitable for the robust target tracking. However, the minimax filter only considers the tracking performance in the receiver. in order to improve the robust tracking performance better, the transmitter waveform could be considered for "smart" target, and then, we will deliberate the waveform selection for robust target tracking

*3.2. Waveform Selection.* According to the review of the control approach in [13], the minimax filter is related to the waveform parameter by the measurement noise covariance. So considering the waveform parameter, (27) could be rewritten as

$$\widehat{\mathbf{x}}_{k+1/k} \left( \boldsymbol{\theta}_{k} \right) = \mathbf{F} \widehat{\mathbf{x}}_{k/k-1} + \mathbf{K}_{k}^{*} \left( \boldsymbol{\theta}_{k} \right) \left( \mathbf{y}_{k} - \mathbf{H} \widehat{\mathbf{x}}_{k/k-1} \right),$$

$$\mathbf{P}_{k+1/k} \left( \boldsymbol{\theta}_{k} \right) = \mathbf{F} \boldsymbol{\Sigma}_{k} \left( \boldsymbol{\theta}_{k} \right) \mathbf{F}^{T} + \mathbf{G} \mathbf{Q} \mathbf{G}^{T},$$

$$\boldsymbol{\Sigma}_{k}^{-1} \left( \boldsymbol{\theta}_{k} \right) = \mathbf{P}_{k/k-1}^{-1} + \mathbf{H}^{T} \left( \mathbf{N}^{-1} \left( \boldsymbol{\theta}_{k} \right) - \mathbf{R}^{-1} \right) \mathbf{H}, \qquad (28)$$

$$\mathbf{K}_{k}^{*} \left( \boldsymbol{\theta}_{k} \right) = \mathbf{F} \boldsymbol{\Sigma}_{k} \left( \boldsymbol{\theta}_{k} \right) \mathbf{H}^{T} \mathbf{N}^{-1} \left( \boldsymbol{\theta}_{k} \right),$$

$$\mathbf{L}_{k}^{*} \left( \boldsymbol{\theta}_{k} \right) = \mathbf{F} \boldsymbol{\Sigma}_{k} \left( \boldsymbol{\theta}_{k} \right) \mathbf{H}^{T} \mathbf{R}^{-1},$$

where  $\theta_k$  is the waveform parameter vector. The minimax filter in (28) contains the transmitted waveform parameters, which is similar to the results of (27) when the transmitted waveform parameter  $\theta_k$  is fixed. However, it does not indicate what is the relationship between the next step transmitted waveform and corresponding tracking performance. Equation (27) just found that the current waveform parameter impects the next step prediction error.

Our aim is to find the relationship between the transmitted waveforms and its corresponding tracking performance, build the optimization criterion, and select the optimal waveform to improve the robust tracking performance better compared to the minimax filter.

Like standard Kalman filter

$$\widehat{\mathbf{x}}_{k/k} = \widehat{\mathbf{x}}_{k/k-1} + \mathbf{G}_{k/k-1} \left( \mathbf{y}_k - \mathbf{H} \widehat{\mathbf{x}}_{k/k-1} \right), \quad (29)$$

$$\widehat{\mathbf{x}}_{k+1/k} = \mathbf{F}\widehat{\mathbf{x}}_{k/k},\tag{30}$$

where  $\hat{\mathbf{x}}_{k/k}$  is the estimated state and  $\mathbf{G}_{k/k-1}$  is the filter gain. Substituting (29) into (30), we have

$$\widehat{\mathbf{x}}_{k+1/k} = \mathbf{F}\widehat{\mathbf{x}}_{k/k-1} + \mathbf{F}\mathbf{G}_{k/k-1}\left(\mathbf{y}\mathbf{H}\widehat{\mathbf{x}}_{k/k-1}\right).$$
(31)

Compared to (13), the relationship between the two gains, **G** and **K**, is

$$\mathbf{G}_{k/k-1} = \mathbf{F}^{-1}\mathbf{K}_k. \tag{32}$$

The estimated state error is defined by

$$\mathbf{e}_{k+1/k+1} = \widehat{\mathbf{x}}_{k+1} - \widehat{\mathbf{x}}_{k+1/k+1}.$$
(33)

According to (11), (29) and (33), the estimated state error could be derived by

$$\mathbf{e}_{k+1/k+1} = \left(\mathbf{I} - \mathbf{G}_{k+1/k}\mathbf{H}\right)\mathbf{e}_{k+1/k} - \mathbf{G}_{k+1/k}\mathbf{n}_{k+1}, \qquad (34)$$

where I is the identity matrix.

The covariance of the estimated state error is defined by

$$\mathbf{P}_{k+1/k+1} = E\left(\mathbf{e}_{k+1/k+1}\mathbf{e}_{k+1/k+1}^{T}\right).$$
(35)

Substituting (34) into (35) and considering the transmitted waveform parameter, we have

$$\mathbf{P}_{k+1/k+1} \left( \boldsymbol{\theta}_{k+1} \right) = \mathbf{M}_{k+1/k} \mathbf{P}_{k+1/k}^{K} \mathbf{M}_{k+1/k}^{T} + \mathbf{M}_{k+1/k} \mathbf{P}_{k+1/k}^{L} \mathbf{M}_{k+1/k}^{T}$$

$$+ \mathbf{G}_{k+1/k} \mathbf{N} \left( \boldsymbol{\theta}_{k+1} \right) \mathbf{G}_{k+1/k}^{T},$$
(36)

where  $M_{k+1/k} = (I - G_{k+1/k}H)$ .

Now, the relationship between the next transmitted waveform parameter and its corresponding target tracking performance is built. Thus, the optimization criterion is to select one transmitted waveform parameter from the parameter database to minimize the trace of the estimated state covariance, that is

$$\boldsymbol{\theta}_{k+1}^* = \min_{\boldsymbol{\theta}_{k+1} \in \Theta} \operatorname{tr} \left\{ \mathbf{P}_{k+1/k+1} \left( \boldsymbol{\theta}_{k+1} \right) \right\}.$$
(37)

When the next optimal transmitted waveform parameter is selected, the measurement noise covariance matrix is known, which improves the robust tracking performance.

The proposed optimal waveform selection can be summarized as follows.

When the minimax filter gets the target state information at time *k* through (28), compute the gain G<sub>k+1/k</sub> through (32).

- (2) Compute the measurement covariance matrix N(θ<sub>k+1</sub>) through (6) for every waveform parameter stored in the database.
- (3) According to (36), compute the trace of every estimation error covariance for every waveform parameter and find the minimization of the values. The waveform parameter, which is corresponding to the minimization value, is the optimal selection for next transmitted waveform.

### 4. Simulation Results

The proposed method is examined in this section. The discrete linear time-invariant system matrices are followed by

$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$
$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$
$$\mathbf{Q} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix},$$
$$\mathbf{S} = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}.$$

The normal target trajectory is as follows:

$$x = 100 + 0.2t + 0.1t^{2} + \cos(5\pi t).$$
(39)

Considering the adversary disturbance, the intelligent target model is

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + 0.08\left(\mathbf{x}_k + \widehat{\mathbf{x}}_{k/k-1}\right).$$
(40)

Some simulation parameters are adopted from [13], the carrier frequency  $\omega_c$  is 25 kHz, and the speed of the transmitted signal *c* is 1500 m/s. the return pulse signal-to-noise ratio  $\eta$  is modeled by

$$\eta = \left(\frac{1000}{r}\right)^4 \eta_{1000},\tag{41}$$

where  $\eta$  is the returned pulse signaltonoise for a target at 1000 m.  $\eta_{1000} \approx 0$  dB. *r* is the target range. The triangularshaped pulse belongs to amplitude-only modulation that is used as transmitted pulse. The waveform parameter is the wavelength  $\lambda$ . We set the parameter database as follows:

$$\lambda = [0.1:0.05:0.3], \tag{42}$$



FIGURE 1: Estimation of the intelligent target by the minimax filter with selected waveform.



FIGURE 2: Estimation of the intelligent target by the minimax filter with fixed waveform.

where 0.05 is the step length. The relationship between the wavelength  $\lambda$  and the measurement noise covariance is [13]

$$R(\lambda_{k+1}) = \begin{bmatrix} \frac{c^2 \lambda_{k+1}^2}{(12\eta)} & 0\\ 0 & \frac{5c^2}{(2\omega_c^2 \lambda_{k+1}^2 \eta)} \end{bmatrix}.$$
 (43)

Firstly, the minimax filters with selected waveform and fixed waveform are used to estimate the intelligent target, respectively. The estimation trajectory of the minimax filter with selected waveform and the target trajectory are shown in Figure 1. The estimation trajectory of the minimax filter with fixed waveform and the target trajectory are shown in Figure 2. It can be seen that two filters can overcome the adversary noise that deliberately maximizes the estimation error and estimate the true target trajectory well.

The performance of the two minimax filters with selected waveform and fixed waveform is shown in Figures 3 and 4. Figure 3 shows the the pulse length is selected in every time in the minimax filter with selected waveform. Figure 4 shows







FIGURE 4: Estimation errors.

the position errors in the two different minimax filters. It can be seen that sometimes, the minimax filter with selected waveform selects the parameters which are the same as the one with fixed waveform. the position errors by the two filters are almost the same. While, in other times, the position error by the minimax filter with selected waveform is smaller than the one by the minimax filter with fixed waveform, since the measurement noise covariance is impacted by the transmitted waveform parameter. When the target range is known, the measurement noise covariance is only changed by the different waveform parameters. The system will select the "best" waveform in order to minimize the the trace of the estimated state covariance. however, the minimax filter with fixed waveform generates the fixed trace of the estimated state covariance when the target range is know

### 5. Conclusion

This paper focuses on the optimal waveform selection for robust target tracking. When the target is assumed to have the "smart" ability, which could increase the estimation error and degrade the target tracking performance, the minimax filter based on the game theory could address the robust tracking problem well from the receiver. On this basis, we improve the minimax filter combining with the waveform selection from the transmitter and derive its the estimation error covariance. Then, according to the relationship between the waveform parameter and measurement noise covariance, the estimation error covariance is related to waveform parameter. Build the optimization criterion that minimizes the estimation error covariance by selecting the waveform parameter at every transmission. The simulation results show the proposed method make the performance of the robust target tracking better than the minimax filter with fixed waveform based game theory.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61004052 and no. 61104005), the Natural Science Foundation of Hebei Province (no. F2013501075), the Fundamental Research Funds for the Central Universities (no. N110323005) and the Doctoral Scientific Research Foundation of Liaoning Province (no. 20131030).

### References

- S. Haykin, "Cognitive radar: a way of the future," *IEEE Signal Processing Magazine*, vol. 23, no. 1, pp. 30–40, 2006.
- [2] S. Haykin, "Cognition is the key to the next generation of radar systems," in *Proceedings of the 13th IEEE Digital Signal Processing* Workshop and 5th IEEE Signal Processing Education Workshop (DSP/SPE '09), pp. 463–467, January 2009.
- [3] S. U. Pillai, D. C. Youla, H. S. Oh, and J. R. Guerci, "Optimum transmit-receiver design in the presence of signal-dependent interference and channel noise," in *Proceedings of the 33rd Asilomar Conference on Signals, Systems, and Computers*, vol. 2, pp. 870–875, Pacific Grove, Calif, USA, 1999.
- [4] S. U. Pillai, H. S. Oh, D. C. Youla, and J. R. Guerci, "Optimum transmit-receiver design in the presence of signal-dependent interference and channel noise," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 577–584, 2000.
- [5] J. R. Guerci and S. U. Pillai, "Theory and application of optimum transmit-receive radar," in *Proceedings of the IEEE International Radar Conference*, pp. 705–710, Alexandria, Va, USA, May 2000.
- [6] J. R. Guerci and S. U. Pillai, "Adaptive transmission radar: the next "wave"?" in *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON '00)*, pp. 779–786, Dayton, Ohio, USA, October 2000.
- [7] M. R. Bell, "Information theory and radar waveform design," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1578–1597, 1993.
- [8] S. M. Sowelam and A. H. Tewfik, "Waveform selection in radar target classification," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 1014–1029, 2000.
- [9] S. Kay, "Waveform design for multistatic radar dsetection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1153–1166, 2009.
- [10] N. A. Goodman, P. R. Venkata, and M. A. Neifeld, "Adaptive waveform design and sequential hypothesis testing for target recognition with active sensors," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 1, pp. 105–113, 2007.

- [11] R. A. Romero, J. Bae, and N. A. Goodman, "Theory and application of SNR and mutual information matched illumination waveforms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 912–927, 2011.
- [12] Y. Wei, H. Meng, Y. Liu, and X. Wang, "Radar phase-coded waveform design for extended target recognition under detection constraints," in *Proceedings of the IEEE Radar Conference*, pp. 1074–1079, May 2011.
- [13] D. J. Kershaw and R. J. Evans, "Optimal waveform selection for tracking systems," *IEEE Transactions on Information Theory*, vol. 40, no. 5, pp. 1536–1550, 1994.
- [14] S. D. Howard, S. Suvorova, and A. Nehorai, "Waveform libraries for radar tracking applications," in *Proceedings of the International Conference on Waveform Diversity and Design*, pp. 1–5, Edinburgh, UK, November 2004.
- [15] S. Suvorova, S. D. Howard, W. Moran, and R. J. Evans, "Waveform libraries for radar tracking applications: maneuvering targets," in *Proceedings of the 40th Annual Conference on Information Sciences and Systems (CISS '06)*, pp. 1424–1428, Princeton, NJ, USA, March 2006.
- [16] D. J. Kershaw and R. J. Evans, "Waveform selective probalilistic data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, pp. 1180–1188, 1997.
- [17] S. P. Sira, A. Papandreou-Suppappola, and D. Morrell, "Dynamic configuration of time-varying waveforms for agile sensing and tracking in clutter," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3207–3217, 2007.
- [18] D. Gu, "A game theory approach to target tracking in sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 41, no. 1, pp. 2–13, 2011.
- [19] D. Simon, "A game theory approach to constrained minimax state estimation," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 405–412, 2006.

### **Research** Article

# **Estimating Time-Varying Beta of Price Limits and Its Applications in China Stock Market**

### Rongquan Bai,<sup>1</sup> Zuoquan Zhang,<sup>1</sup> and Menggang Li<sup>2</sup>

<sup>1</sup> School of Science, Beijing Jiaotong University, Beijing 100044, China <sup>2</sup> China Center for Industrial Security Research, Beijing 100044, China

Correspondence should be addressed to Zuoquan Zhang; zqzhang@bjtu.edu.cn

Received 6 December 2012; Accepted 16 June 2013

Academic Editor: Rung Ching Chen

Copyright © 2013 Rongquan Bai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an estimation method of time-varying beta of price limits. It uses China stock market trading data to estimate time-varying beta and researches on systemic risk in China stock market. By comparing prediction errors of market model, SS market model, and Censored-SS market model, it verifies the effectiveness of Censored-SS market model. Furthermore it has some meaningful conclusions in China stock market.

### 1. Introduction

Since the foundation of the Capital Asset Pricing Model (CAPM), many researchers focus a great deal of attention on estimating the beta. The most common method of estimating systemic risk is to estimate the slope of CAPM. It is actually a constant and does not vary with time.

In recent years, a growing number of researches have shown that systemic risk of portfolio is unstable, and it will vary with time. There are some works in the literature about estimations of time-varying beta. M-GARCH method, Kalman filtering method, and time-varying market model method proposed by Schwert and Seguin are the most common methods [1].

M-GARCH method was first introduced by Bollerslev [2]. It uses the conditional variance and conditional covariance generated by the multivariate GARCH model to build time-varying beta sequence. Many studies, such as Brooks et al., have applied this method [3–6].

Another method is kalman filtering. It assumes that beta is subjected to a random process firstly. And then it estimates the time-varying beta and generates the relevant parameters of the market model recursively from the initial value. Groenewold and Fraser have studied this method [7].

The third method is extended market model method proposed by Schwert and Seguin. They introduce the

heteroscedasticity of comprehensive market volatility and time-varying characteristics of beta to the market model. And the time-varying beta sequence is calculated by estimating the model's parameters. The relevant researches have Episcopos, Reyes, and so on [8, 9].

Brooks et al. make a comparison of the three methods in the Australian market. They find that the results of the three methods are significantly different. By comparing the MSEs of estimating betas they believe that kalman filtering is the best method in Australian stock market [10].

Although there have been several estimating methods of time-varying beta and empirical researches of these methods in the literaturs, institutional factors such as price limits' effects on estimation of time-varying beta have not been taken into account. Price limits that restrics daily highest change rate of the stock market are introduced as a mechanism to stabilize the market. Many countries and areas in the world have this restriction, such as China, Australia, France, Japan, and Korea. However, as daily price fluctuation is limited within a predetermined range, it will make the observed return deviate from balanced return and change the random characteristics of return sequence; worse still, instead of using balanced return, using observed return to do the regression estimation leads to a biased estimation of model parameters naturally [11]. So these methods are not suitable for estimating beta of price limits [12-19].
Our research will be based on the extended market model proposed and censored model. We call it Censored-SS market model.

#### 2. Censored-SS Market Model

2.1. Market Model. Traditional market model is:

$$R_{i,t} = \alpha_i + \beta_i R_{m,t} + \varepsilon_{i,t},\tag{1}$$

where  $R_{i,t}$  is the return of portfolio *i* at *t*;  $R_{m,t}$  is the return of comprehensive market at *t*; the slope  $\beta_i$  of regression model is the beta of portfolio *i*, and it represents the systemic risk of portfolio *i*.

The estimation of beta in (1) is

$$\widehat{\beta}_i = \frac{\operatorname{cov}\left(R_{i,t}, R_{m,t}\right)}{\sigma_m^2},\tag{2}$$

where the numerator is the covariance between portfolio's return and comprehensive market's return, and the denominator is the variance of comprehensive market' return. So the estimation of beta based on market model is a constant actually, and it will not vary with time.

2.2. Censored Regression Model. Censored regression model is a kind of limited dependent variable models [12]. Censored refers to the observed values of samples in a range that are used by a same value. When regression model is censored, the model is a censored model. Its expression is

$$y_{i}^{*} = x_{i}^{\prime}\beta + \varepsilon_{i},$$

$$y_{i} = \begin{cases} c_{1}, & y_{i}^{*} \le c_{1}, \\ y_{i}^{*}, & c_{1} < y_{i}^{*} < c_{2}, \\ c_{2}, & y_{i}^{*} \ge c_{2}, \end{cases}$$
(3)

where  $y_i^*$  is the real dependent variable,  $y_i$  is the censored dependent variable,  $x_i$  is the independent variable matrix,  $\beta$  is the parametric vector, and  $\varepsilon_i$  is the disturbance term.  $c_1$  and  $c_2$  are two fixed values, and they represent the left and right censored points.

It generally uses the maximum likelihood estimation to estimate (3). Its log-likelihood function is

$$L(\beta, \sigma) = \sum_{y_i^* \le c_1} \log F\left(\frac{c_1 - x_i'\beta}{\sigma}\right) + \sum_{c_1 < y_i < c_2} \log f\left(\frac{y_i - x_i'\beta}{\sigma}\right) + \sum_{y_i^* \ge c_2} \log \left[1 - F\left(\frac{c_2 - x_i'\beta}{\sigma}\right)\right],$$
(4)

where F is the cumulative distribution function of the disturbance item and f is the density function of the disturbance item.

*2.3. SS Market Model.* Schwert and Seguin proposed a market model of time-varying beta based on market model (namely SS market model) [1]. It is defined as

$$R_{i,t} = \alpha_i + \beta_{i,t} R_{m,t} + \varepsilon_{i,t},\tag{5}$$

where beta is a variable changing with time. And it can be divided into two parts:

$$\beta_{i,t} = \beta_i + \frac{\delta_i}{\sigma_{m,t}^2},\tag{6}$$

where  $\beta_i$  and  $\delta_i$  are constants and  $\sigma_{m,t}^2$  is the conditional variance of comprehensive market return. Equation (6) shows the relationship between systemic risk and comprehensive market fluctuation. If  $\delta_i > 0$ , systemic risk and comprehensive market fluctuation have a negative relation, and if  $\delta_i < 0$ , systemic risk and comprehensive market fluctuation have a positive relation.

Substituting (6) into (5),

$$R_{i,t} = \alpha_i + \beta_i R_{m,t} + \delta_i \frac{R_{m,t}}{\sigma_{m,t}^2} + \varepsilon_{i,t}.$$
 (7)

This is the SS market model, and three are three parameters to be estimated:  $\alpha_i$ ,  $\beta_i$ , and  $\delta_i$ .  $\sigma_{m,t}^2$  can be estimated by GARCH model of comprehensive market return, and its estimation can be obtained,  $\hat{\sigma}_{m,t}^2$ . So the actual estimating equation is

$$R_{i,t} = \alpha_i + \beta_i R_{m,t} + \delta_i \frac{R_{m,t}}{\hat{\sigma}_{m,t}^2} + \varepsilon_{i,t}.$$
 (8)

Equation (8) shows that Schwert and Seguin add timevarying characteristic of beta and heteroscedasticity characteristic of comprehensive market return added in traditional market model actually.

2.4. Return Characteristics of Price Limits. As many countries implement the price limits, there have been some works in the literature that research on return characteristics of price limits; see Wei et al. [12–19]. Market trading price's fluctuation range is generally calculated on the basis of the previous day's closing price.  $P_{t-1}$  represents the closing price of trading day t - 1 that we can observe. And limited rates of price rises and falls are  $l_u$  and  $l_d$ , respectively. So the closing price of trading day t that we can observe will fall into interval  $[P_{t-1}(1 - l_d), P_{t-1}(1 + l_u)]$ . That is to say, when stock price rises to its daily limit or stock price falls to its daily limit, the balanced closing price  $P_t^*$  will exceed this interval and  $P_t^*$ cannot be observed [20]. The relationship between observed price  $P_t$  and balanced price  $P_t^*$  can be described as

$$P_{t} = \begin{cases} P_{t-1} (1 + l_{u}), & P_{t}^{*} \geq P_{t-1} (1 + l_{u}) \\ P_{t}^{*}, & P_{t-1} (1 - l_{d}) < P_{t}^{*} < P_{t-1} (1 + l_{d}) \\ P_{t-1} (1 - l_{d}), & P_{t}^{*} \leq P_{t-1} (1 - l_{d}). \end{cases}$$

$$\tag{9}$$

Equation (9) demonstrates that the observed price will be equal to the balanced price only when the balanced price lies in the predetermined range. Equation (9) is divided by  $P_{t-1}$  and we take the log of it as follows:

$$R_{t} = \begin{cases} L_{u}, & \log\left(\frac{P_{t}^{*}}{P_{t-1}}\right) \geq L_{u} \\ \log\left(\frac{P_{t}^{*}}{P_{t-1}}\right), & L_{d} < \log\left(\frac{P_{t}^{*}}{P_{t-1}}\right) < L_{u} \\ L_{d}, & \log\left(\frac{P_{t}^{*}}{L_{d}}\right), \end{cases}$$
(10)

where  $R_t = \log(P_t/P_{t-1})$ , and it is the observed return;  $L_u = \log(1 + l_u)$ ;  $L_d = \log(1 - l_d)$ .

 $\log(P_t^*/P_{t-1})$  in (10) can be divided as follows:

$$\log\left(\frac{P_{t}^{*}}{P_{t-1}}\right) = \log\left(\frac{P_{t}^{*}}{P_{t-1}^{*}}\right) + \log\left(\frac{P_{t-1}^{*}}{P_{t-1}}\right)$$

$$= R_{t}^{*} + LO_{t-1},$$
(11)

where  $R_t^*$  is the balanced return of trading day t;  $LO_{t-1}$  is called residual return, and it is equal to  $\log(P_{t-1}^*/P_{t-1})$ . When the closing price of trading day t - 1 does not reach the price limits, because of  $P_{t-1}^* = P_{t-1}$ ,  $LO_{t-1}$  will be equal to 0. When the closing price of trading day t - 1 reaches the price limits, because of  $P_{t-1}^* \neq P_{t-1}$ ,  $LO_{t-1}$  will not be equal to 0, and it demonstrates that the unrealized return of trading day t - 1 caused by price limits transfers to the next trading day. We substitute (11) into (10) as follows:

$$R_{t} = \begin{cases} L_{u}, & R_{t}^{*} + LO_{t-1} \ge L_{u} \\ R_{t}^{*} + LO_{t-1}, & L_{d} < R_{t}^{*} + LO_{t-1} < L_{u} \\ L_{d}, & R_{t}^{*} + LO_{t-1} \le L_{d}. \end{cases}$$
(12)

Equation (12) is the censored law of price limits. It demonstrates that when the sum of balanced return of trading day t and residual return of trading day t - 1 is not in  $(L_d, L_u)$  the observed return will be censored.

2.5. SS Market Model Expansion: Censored-SS Market Model. When stock price of trading day t - 1 does not reach the price limits,  $LO_{t-1} = 0$ . And (12) becomes

$$R_{t} = \begin{cases} L_{u}, & R_{t}^{*} \ge L_{u} \\ R_{t}^{*}, & L_{d} < R_{t}^{*} < L_{u} \\ L_{d}, & R_{t}^{*} \le L_{d}. \end{cases}$$
(13)

Like (12), (13) is a censored law. However, when stock price trading day t - 1 reaches the price limits,  $LO_{t-1} \neq 0$ , and it will transfer to the next trading day's observed return. It makes the return's censored structure of the latter trading day complicated, as  $LO_{t-1}$  cannot be observed.

We cannot apply traditional regression model to the observed return in the censored structure of price limits. In order to apply the censored law to the observed return, we borrow the method of Chen and Jau-Lian [11, 12] that deletes the return data of the trading day after the price reaches the price limits. Then the remaining data are the accordance with the censored law.

So we introduce the censored model into the SS market model and we can get a new model. We call it Censored-SS market model. The model is

$$R_{i,t}^* = \alpha_i + \beta_i R_{m,t} + \delta_i \frac{R_{m,t}}{\widehat{\sigma}_{m,t}^2} + \varepsilon_{i,t}, \qquad (14)$$

where,  $R_{i,t}^*$  is the implicit dependent variable in censored model. It is subject to the following equations:

$$R_{i,t} = \begin{cases} L_u, & R_{i,t}^* \ge L_u \\ R_{i,t}^*, & L_d < R_{i,t}^* < L_u \\ L_d, & R_{i,t}^* \le L_d. \end{cases}$$
(15)

So the estimation of (14) is to estimate the SS model. When it is assumed that the error term  $\varepsilon_{i,t}$  is subject to a normal distribution, the method to estimate this model is generally maximum likelihood method. Its likelihood function is

$$\begin{split} L\left(\alpha_{i},\beta_{i},\delta_{i},\frac{\sigma_{\varepsilon_{i}^{2}}}{R_{i,t}},\exists t\in H_{n}\cup H_{u}\cup H_{d}\right)\\ &=\prod_{t\in H_{n}}\frac{1}{\sigma_{\varepsilon_{i}}}\phi\left(\frac{R_{i,t}-\alpha_{i}-\beta_{i}R_{m,t}-\delta_{i}\left(R_{m,t}/\widehat{\sigma}_{m,t}^{2}\right)}{\sigma_{\varepsilon_{i}}}\right)\\ &\times\prod_{t\in H_{u}}\frac{1}{\sigma_{\varepsilon_{i}}}\Phi\left(\frac{L_{u}-\alpha_{i}-\beta_{i}R_{m,t}-\delta_{i}\left(R_{m,t}/\widehat{\sigma}_{m,t^{2}}\right)}{\sigma_{\varepsilon_{i}}}\right)\\ &\times\prod_{t\in H_{d}}\frac{1}{\sigma_{\varepsilon_{i}}}\left[1-\Phi\left(\frac{L_{d}-\alpha_{i}-\beta_{i}R_{m,t}-\delta_{i}\left(R_{m,t}/\widehat{\sigma}_{m,t}^{2}\right)}{\sigma_{\varepsilon_{i}}}\right)\right], \end{split}$$
(16)

where  $\phi(\cdot)$  and  $\Phi(\cdot)$  represent the probability density function and cumulative distribution function of standard normal distribution, respectively.  $\sigma_{e_i^2}$  is the variance of disturbance term.  $H_n$  represents the collection of the trading day that does not reach price limits and trading day before does not reach too.  $H_u$  represents the collection of trading day that rises to its price limit, but the trading day before does not reach.  $H_d$ represents the collection of trading day that falls to its price limits, but the trading day before does not reach. The union of  $H_n$ ,  $H_u$ , and  $H_d$  is the collection of trading days of which the day before does not reach its price limits.



FIGURE 1: The histogram and statistics of  $R_{mt}$ .

#### 3. Estimation of Time-Varying Beta of Price Limits

3.1. China Stock Market Background. Now Shanghai and Shenzhen stock exchanges adopt automatic computer exchange system. There are five trading days from Monday to Friday and holidays are excepted. Each trading day is cut off by morning market and afternoon market. Morning market is from 9:30 to 11:30, and afternoon market is from 13:00 to 15:00. From 9:15 to 9:26 is called auction time before morning market, and traders can submit bills, but the bills cannot be completed. Until 9:25, the host computer starts and generates opening prices according to the principal of price and time preference. It uses continuous auction after 9:30. The closing price in Shanghai stock exchange is the average weighted price of all transactions during the last minute. If there are no trades during the day, the closing price is the former trading day's closing price. The closing price in Shenzhen stock exchange is generated by call auction. If call auction cannot generate closing price, the closing price is the average weighted price of all trades during the last minute. If there are no trades during the day, the closing price is the former trading day's closing price [20].

China stock market has daily price limits since 1996. The fluctuation of stock price in a trading day is less than 10% except that this trading day is the stock's first trading day.

3.2. Data and Sample Selection. All the trading data in this paper are from CSMAR database. The selected period is from the beginning of 2009 to the end of 2011. There are a total of 730 trading days. The comprehensive market return refers to the Shanghai composite index return. The stock objects in this research are limited to the A-shares listed on the Shanghai stock exchange. In order to ensure that all the stock objects have a long enough trading data, we remove the stocks listed after 2009. We classify stocks into large corporations and small corporations and use time-varying beta to research on systemic risk in China stock market's large corporation and small corporations significantly, this paper sorts the stocks in Shanghai stock exchange according to the total value of trade and selects 30 large corporations and 30 small corporations.

TABLE 1:  $R_{m,t}$ 's autocorrelation coefficients and partial correlation coefficients.

Level	Autocorrelation coefficient	Partial correlation coefficient	Q-statistics	Probability
1	0.042	0.042	1.3213	0.250
2	-0.000	-0.002	1.3213	0.517
3	0.026	0.027	1.8370	0.607
4	-0.032	-0.034	2.5841	0.630
5	-0.015	-0.012	2.7425	0.740
6	-0.007	-0.007	2.7784	0.836
7	0.045	0.048	4.2915	0.746
8	0.012	0.008	4.4049	0.819
9	0.049	0.048	6.1816	0.722
10	0.044	0.037	7.6465	0.663

Then we select 10 stocks from them, respectively, which have the most time of reaching price limits as samples.

3.3. Simulation of Market Return's Conditional Variance. The histogram and statistics of the Shanghai composite index's return sequence  $R_{m,t}$  from 2009 to 2011 are shown in Figure 1. The skewness is -0.484593. Its kurtosis is 4.691811. Its Jarque-Bera value is 115.6304. It means that the distribution of China's stock comprehensive market return is not normal. It is leptokurtic and left skewed. The Shanghai composite index return sequence  $R_{m,t}$ 's autocorrelation coefficients and partial correlation coefficients are shown in Table 1. From Table 1 we can find that  $R_{m,t}$  does not have serial correlation. So we can fit GARCH models whose conditional mean equation is  $R_{m,t} = \mu + \varepsilon_t$ .

We choose GARCH-M as the autoregressive conditional heteroscedasticity model of comprehensive market return [2].

GARCH-M model's ARCH coefficient  $a_1$  is 0.037161. GARCH coefficient  $a_2$  is 0.940227.  $a_1 + a_2 = 0.977338$ . It means that persistence of volatility is 0.977338. Namely, residual impact still has 0.977338<sup>5</sup> or 89% after five trading days.

Mean	Median	Maximum	Minimum	Standard deviation	Size
0.000273	0.000216	0.001230	0.000133	0.000157	730



FIGURE 2: The conditional variance  $\hat{\sigma}_{m,t}^2$  of comprehensive market return.

Using the selected GARCH-M model we can fit the conditional variance of comprehensive market return within samples. The fitting result of  $\hat{\sigma}_{m,t}^2$  sequence is shown in Figure 2. The conditional variance during 2009 is greater, as shown in Figure 2. It means that the comprehensive market has a sharp fluctuation in this period [20]. The descriptive statistics of  $\hat{\sigma}_{m,t}^2$  is shown in Table 2.

3.4. Estimation Results. We put the Shanghai composite index return and conditional variance's estimates sequence into (14). And we make use of the large corporations and small corporations' return data to estimate Censored-SS market model, and the results are shown in Table 3. After acquiring each corporation's estimations of  $\beta_i$  and  $\delta_i$ , we put them into (6). Thus we get the estimated sequence of each stock's time-varying beta of price limits. The descriptive statistics is shown in Tables 4 and 5.

#### 4. Research on Systemic Risk in China Stock Market

According the results of time-varying beta estimated by Censored-SS model, systemic risk in China stock market is analyzed from two aspects that are corporation size and market fluctuation.

4.1. Corporation Size. Tables 4 and 5 show the beta's statistics of large corporations and small corporations, respectively. We use large corporations' beta mean sequences and small corporations' beta mean sequences to do the nonparameter wilcoxon matched pair rank test, and the z value and level of significance are 0.7939 and 0.4274, respectively. So corporation size has no effects on the mean of systemic risk significantly.

We use large corporations' beta standard deviation sequences and small corporations' beta standard deviation

Number	$lpha_i$	$\beta_i$	$\delta_i$		
		Large corporation	18		
1	7.42E - 05	0.998446	-5.18E - 06		
2	-0.001022	1.045616	-4.26E - 05		
3	-0.000279	0.754530	2.21E - 05		
4	-0.000620	1.010184	6.20E - 05		
5	0.001757	0.964994	5.06E - 05		
6	8.00E - 05	1.078617	2.87E - 05		
7	6.72E - 05	0.949563	4.01E - 05		
8	-0.000402	1.434350	-4.71E - 05		
9	0.000412	1.071771	1.92E - 05		
10	-0.000868	1.009653	-2.57E - 05		
		Small corporation	ns		
1	0.000853	1.764856	-0.000210		
2	0.000320	1.277261	-2.71E - 05		
3	0.001859	1.240980	-1.85E - 05		
4	0.002455	1.241941	-7.61E - 05		
5	0.000430	0.766067	4.56E - 05		
6	0.001145	1.028433	6.92E - 06		
7	0.000543	1.277040	-0.000105		
8	0.000659	1.077382	6.54E - 06		
9	0.000725	1.340762	-1.78E - 05		
10	0.000209	1.191594	-2.47E - 05		

TABLE 3. Results of Censored-SS model

sequences to do the nonparameter wilcoxon matched pair rank test, and the z value and level of significance are 0.1134 and 0.9097, respectively. So corporation size has no effects on the fluctuation of systemic risk significantly.

4.2. Market Fluctuation.  $\delta_i$  represents the relationship between systemic risk and market fluctuation. If  $\delta_i > 0$ , systemic risk and market fluctuation have a negative relation, and if  $\delta_i < 0$ , systemic risk and market fluctuation have a positive relation. Table 6 shows the statistics of  $\delta_i$ . From Table 6 we can see that there is only a slight difference between large corporations'  $|\delta_i|$  and small corporations'  $|\delta_i|$ . We use large corporations'  $|\delta_i|$  sequences and small corporations'  $|\delta_i|$  sequences to do the nonparameter wilcoxon matched pair rank test, and the *z* value and level of significance are 0.1134 and 0.9097, respectively. So market fluctuation's effect on the absolute level of systemic risk has no difference between different corporation sizes.

From Table 6 we can also see that the mean of large corporations'  $\delta_i$  is positive and the mean of small corporations'  $\delta_i$  is negative. And six stocks of large corporations have positive  $\delta_i$  and four stocks of large corporations have negative  $\delta_i$ . However, three stocks of large corporations have positive  $\delta_i$  and seven stocks of large corporations have

TABLE 4: The descriptive statistics of time-varying beta of large corporations.

Number	Mean	Maximum	Minimum	Standard deviation
1	0.975618	0.994234	0.959502	0.007778
2	0.857880	1.010975	0.725347	0.063969
3	0.851923	0.920679	0.772501	0.033186
4	1.283414	1.476304	1.060600	0.093101
5	1.187985	1.345408	1.006140	0.075982
6	1.205096	1.294385	1.101955	0.043097
7	1.126281	1.251037	0.982171	0.060215
8	1.226783	1.396050	1.080249	0.070727
9	1.156384	1.216118	1.087384	0.028831
10	0.896395	0.988755	0.816439	0.038592
Total	1.076776	1.476304	0.725347	0.166302

TABLE 5: The descriptive statistics of time-varying beta of small corporations.

Number	Mean	Maximum	Minimum	Standard deviation
1	0.839398	1.594092	0.186064	0.315097
2	1.157833	1.255224	1.073522	0.040694
3	1.159452	1.225937	1.101896	0.027780
4	0.906573	1.180060	0.669817	0.114274
5	0.967024	1.108891	0.803147	0.068474
6	1.058929	1.080458	1.034060	0.010391
7	0.814311	1.191658	0.487644	0.157671
8	1.106203	1.126550	1.126550	0.009821
9	1.262318	1.326288	1.206941	0.026729
10	1.082743	1.171509	1.005898	0.037090
Total	1.035478	1.594092	0.186064	0.185705

TABLE 6: The statistics of  $\delta_i$  estimated by Censored-SS model.

Large corporations	Small corporations
	$\delta_i$
1.02E - 05	-4.20E - 05
6.20E - 05	4.56E - 05
-4.71E - 05	-0.00021
3.85E - 05	7.26E - 05
6	3
4	7
	$\delta_i$
3.43E - 05	4.89E - 05
6.20E - 05	0.00021
5.18E - 06	6.54E - 06
1.71E - 05	6.77E - 05
	Large corporations 1.02E - 05 6.20E - 05 -4.71E - 05 3.85E - 05 6 4 3.43E - 05 6.20E - 05 5.18E - 06 1.71E - 05

negative  $\delta_i$ . Namely, when market fluctuation is increasing, systemic risk of large corporation is decreasing and systemic risk of small corporation is increasing. So When market fluctuation is increasing, investment portfolio transfers to

TABLE 7: The statistics of RMSE within samples.

	Censored-SS market model	SS market model	Market model
Mean	0.026684	0.026684	0.026766
Maximum	0.030186	0.030194	0.030197
Minimum	0.024953	0.024957	0.024963
Standard deviation	0.001592	0.001598	0.001611

large corporation stock can decrease risk in China stock market.

#### 5. Comparison of Market Model, SS Market Model, and Censored-SS Market Model

*5.1. Prediction Errors within Samples.* In order to compare the three models' prediction error within samples, this paper uses the same samples to estimate market model and SS market model. The three models' descriptive statistics of RMSE are shown in Table 7.

Censored-SS model's RMSE is less than that of SS market model. And SS market model's RMSE is less than that of market model. So when considering the time-varying characteristics, SS market model and Censored-SS market model acquire a more accurate return prediction than market model within samples. And when taking price limits into account, Censored-SS market model acquires a more accurate prediction than SS market model.

So for fitting error of stock return within samples, we believe that the beta estimation of Censored-SS model is more accurate than market model and SS market model.

5.2. Beta. We take ten large corporations and ten small corporations as two combinations. And the combination's time-varying beta sequence is equal to the average sequence of the ten corporations' daily beta, respectively, as shown in Figures 3 and 4. The blue line represents the constant beta sequence estimated by market model. The green line represents the beta sequence estimated by SS market model. And the red line represents the beta sequence estimated by Censored-SS market model. These two figures show that the beta sequence estimated by SS market model is less than the beta sequence estimated by Censored-SS market model most of the time. In other words, ignoring the price limits will lead to underestimate the time-varying systemic risk.

#### 6. Conclusion

This paper proposes an estimation method of time-varying beta of price limits and researches on the systemic risk in China stock market. This paper introduces the three main estimation methods of time-varying beta in the literature firstly. And combined with price limits' effects on stock return, it uses a Censored-SS market model. It uses trading data of China's stock market to estimate time-varying beta. Finally by comparing prediction errors of market model, SS market model, and Censored-SS market model, it verifies



FIGURE 3: Beta sequences of large corporations combination.



FIGURE 4: Beta sequences of small corporations combination.

the effectiveness of Censored-SS market model. The main conclusions are as follows.

- Censored-SS market model can provide more accurate beta estimation than market model and SS market model. Moreover, it will lead to an underestimation of systemic risk when taking the time-varying characteristics into account but ignoring the factor of price limits.
- (2) Corporation size has no significant effects on systemic risk's mean and fluctuation in China stock market. However, market fluctuation has different effects on large corporations and small corporation. When market fluctuation increases, large corporation's systemic risk will decrease and small corporation's systemic risk will increase. So the systemic risk gap between large corporations and small corporations will be larger.

#### Acknowledgment

The project is supported by the research on China's industrial security index (B09C11010020).

#### References

- G. W. Schwert and P. J. Seguin, "Heteroskedasticity in stock returns," *Journal of Finance*, vol. 45, pp. 1129–1155, 1990.
- [2] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307– 327, 1986.
- [3] R. D. Brooks, R. W. Faff, and M. Ariff, "An investigation into the extent of beta instability in the Singapore stock market," *Pacific Basin Finance Journal*, vol. 6, no. 1-2, pp. 87–101, 1998.
- [4] F. Lie, R. Brooks, and R. Faff, "Modeling the equity beta risk of Australian financial sector companies," *Australian Economic Papers*, vol. 9, pp. 301–311, 2000.
- [5] M. D. McKenzie, R. D. Brooks, R. W. Faff, and Y. K. Ho, "Exploring the economic rationale of extremes in GARCH generated betas The case of U.S. banks," *Quarterly Review of Economics and Finance*, vol. 40, no. 1, pp. 85–106, 2000.
- [6] M. D. McKenzie, R. D. Brooks, and R. W. Faff, "The use of domestic and world market indexes in the estimation of timevarying betas," *Journal of Multinational Financial Management*, vol. 10, no. 1, pp. 91–106, 2000.
- [7] N. Groenewold and P. Fraser, "Time-varying estimates of CAPM betas," *Mathematics and Computers in Simulation*, vol. 48, no. 4–6, pp. 531–539, 1999.
- [8] A. Episcopos, "Stock return volatility and time-varying betas in the Toronto stock exchange," *Quarterly Journal of Business and Economics*, vol. 35, pp. 28–38, 1996.
- [9] M. G. Reyes, "Size, time-varying beta, and conditional heteroscedasticity in UK stock returns," *Review of Financial Economics*, vol. 8, no. 1, pp. 1–10, 1999.
- [10] R. D. Brooks, R. W. Faff, and M. D. McKenzie, "Time-varying beta risk of australian industry portfolios: a comparison of modelling techniques," *Australian Journal of Management*, vol. 23, no. 1, pp. 1–22, 1998.
- [11] C. Chen and J. Jau-Lian, "The impact of price limits on foreign currency future's price volatility and market efficiency," *Global Finance Journal*, vol. 7, no. 1, pp. 13–25, 1996.
- [12] S. X. Wei, "A censored-GARCH model of asset returns with price limits," *Journal of Empirical Finance*, vol. 9, no. 2, pp. 197– 223, 2002.
- [13] P.-H. Chou, "A Gibbs sampling approach to the estimation of linear regression models under daily price limits," *Pacific Basin Finance Journal*, vol. 5, no. 1, pp. 39–62, 1997.
- [14] P.-H. Chou, "Modeling daily price limits," *International Review of Financial Analysis*, vol. 8, no. 3, pp. 283–301, 1999.
- [15] H.-C. R. Huang, "Tests of regime-switching CAPM under price limits," *International Review of Economics and Finance*, vol. 12, no. 3, pp. 305–326, 2003.
- [16] Y.-S. Huang, T.-W. Fu, and M.-C. Ke, "Daily price limits and stock price behavior: evidence from the Taiwan stock exchange," *International Review of Economics and Finance*, vol. 10, no. 3, pp. 263–288, 2001.

- [17] K. A. Kim, "Price limits and stock market volatility," *Economics Letters*, vol. 71, no. 1, pp. 131–136, 2001.
- [18] K. A. Kim and P. Limpaphayom, "Characteristics of stocks that frequently hit price limits: empirical evidence from Taiwan and Thailand," *Journal of Financial Markets*, vol. 3, no. 3, pp. 315–332, 2000.
- [19] S. B. Lee and D. J. Kim, "Price limits and beta," *Review of Quantitative Finance and Accounting*, vol. 9, no. 1, pp. 35–52, 1997.
- [20] C. Lang-Nan, D. Hong-Wei, D. Hong-Gang, H. Zhao-Xia, and C. Deng-Ta, *Research on Asset Pricing*, China Financial and Economic Publishing House, 2008.

# **Research** Article

# **Qualitative and Quantitative Integrated Modeling for Stochastic Simulation and Optimization**

### Xuefeng Yan,<sup>1</sup> Yong Zhou,<sup>1</sup> Yan Wen,<sup>1</sup> and Xudong Chai<sup>2</sup>

<sup>1</sup> College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China <sup>2</sup> Beijing Simulation Center, Beijing 100854, China

Correspondence should be addressed to Xuefeng Yan; xuefeng.yan@gmail.com

Received 21 April 2013; Accepted 17 May 2013

Academic Editor: Neal N. Xiong

Copyright © 2013 Xuefeng Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The simulation and optimization of an actual physics system are usually constructed based on the stochastic models, which have both qualitative and quantitative characteristics inherently. Most modeling specifications and frameworks find it difficult to describe the qualitative model directly. In order to deal with the expert knowledge, uncertain reasoning, and other qualitative information, a qualitative and quantitative combined modeling specification was proposed based on a hierarchical model structure framework. The new modeling approach is based on a hierarchical model structure which includes the meta-meta model, the meta-model and the high-level model. A description logic system is defined for formal definition and verification of the new modeling specification. A stochastic defense simulation was developed to illustrate how to model the system and optimize the result. The result shows that the proposed method can describe the complex system more comprehensively, and the survival probability of the target is higher by introducing qualitative models into quantitative simulation.

#### 1. Introduction

Stochastic simulation has become a highly effective and essential part of all scientific fields to analyze, reconstruct, and optimize the objective world without the need to perform experiments on a physical product or an actual system. In theoretical and experimental research, it has become another important way to reveal the internal and essential laws of the real world. To study and gain insight into real phenomena, a stochastic model should be constructed for some particular purpose at an appropriate level of abstraction or fidelity.

In the field of stochastic simulation, whenever we mention "qualitative model," the phrase "quantitative model" will naturally come to mind. In fact, "simulation model" generally refers to a quantitative model if not particularly described, and most research is based on the mathematical model [1]. Precise mathematical models are built to describe the system structure and behavior, especially the logic and functionality on the timeline. The simulation is carried out by solving the equations in a numerically calculated fashion. The simulation results rely on the accuracy of the models. However, the mathematical perfection is not representative of the authenticity of the system and the subtle experiential meaning of the real world cannot be modeled by mathematical equations. On the other hand, the objects we studied, such as aircraft, weapons, and space systems, are increasingly complex. This is particularly true of giant, complex system. We can only have or create some of the mathematical models with certain accuracy. It is almost impossible to construct all the quantitative models and complete their Verification, Validation and Accreditation (VVA). Furthermore, not all of the simulation requires a precise mathematical model. For example, sometimes we are only interested in the macroevolution trend of a system, rather than time-specific values.

The symbol qualitative model can contain various forms of information and has reasoning and learning ability. The structure and behavior of the actual system are described in an abstract form, focusing on the causality and not on mathematical equations. It is widely used in many fields associated with physics, chemistry, ecology, biology, fault diagnosis, mechanical manufacturing, industrial systems, and Artificial Intelligence (AI) [2]. We can see that the combination of qualitative and quantitative attributes shows promise for stochastic simulation. Many scholars have made important progresses in this field [3–9]. Due to the direct usage of existing expertise, qualitative and quantitative integrated methods have many significant advantages.

- When it is difficult to build all the quantitative models and the stochastic simulation cannot be constructed because some models are lacking, the qualitative model could be a necessary complement.
- (2) Qualitative modeling is effective for some fields where most of the knowledge is expressed by symbols, language, or graphics directly.
- (3) When we are just interested in the macroevolution or the essential qualitative phenomenon, it is not necessary to occupy a large number of computing time and resources for quantitative simulation.
- (4) The static structure of the simulation can be organized based on qualitative models and at run-time, qualitative models can intelligently choose the better execution branch or data based on the schedule engine. Different Detail of Level (DOL) resolution can be constructed for a system at different abstraction levels.
- (5) The traditional evaluation and optimization can be innovated because the qualitative mode is a part of the simulation and online assessment could be made.

We can see that the qualitative model brings an unprecedented opportunity to improve traditional stochastic simulation. But it also faces with the following challenges.

- (1) There are a large number of different types of qualitative models in different application fields, and the requirements, interfaces, and forms are varied.
- (2) The qualitative modeling methods and symbolic languages are also diverse in different applications fields. These heterogeneous models are incompatible with each other and it is difficult to simulate together.
- (3) The loose and redundancy qualitative models should be integrated with the rigorous quantitative models to form the stochastic simulation with a precise logical structure. Many effects are needed in qualitative and quantitative hybrid simulation engines [10].

There are lots of classic researches in quantitative modeling, such as the specification named Discrete Event Systems Specification (DEVS) for discrete event systems and COllaborative SIMulation (COSIM) for multidisciplinary virtual prototype modeling and simulation [11-16]. In order to deal with expert knowledge, uncertain reasoning, and other qualitative information, a qualitative and quantitative integrated modeling specification and the theoretical framework for stochastic simulation and optimization are significant. In this paper, a hierarchical model structure is proposed, including meta-meta model, meta-model, and the high-level model. The qualitative and quantitative heterogeneous model and integrated relationship were described at a higher abstraction level. The description logic system is defined for the framework based on the formal description and verification of the modeling specification.

The rest of this paper is organized as follows. Section 2 briefly introduces the related researches of qualitative and quantitative modeling for stochastic simulation. In Section 3, a qualitative and quantitative integrated modeling specification is presented, including the modeling framework, description logic system, and formal definitions. Section 4 proves the self-close feature of the models. In Section 5, a qualitative and quantitative mixed stochastic defense system is modeled and simulated. Section 6 draws related conclusions and points out future work.

#### 2. Related Works

2.1. Qualitative Model in Stochastic Simulation. A complex stochastic simulation is always composed of various subsystems. To analyze and optimize the performance, qualitative models have been investigated and applied to more and more fields [17]. In [1], tactical decision making based on fuzzy logic was applied to an underwater vehicle in an engagement-level simulation. A light torpedo and a submarine were modeled based on DEVS and the submarine model calls the fuzzy logic model to conduct a tactical decision. The fuzzy logic was implemented as the Python script tactic description file. By adopting the fuzzy logic, a smoother result was obtained than fixed established tactics and the survival possibility of the submarine was enhanced. SHAO Chen-xi believed that qualitative modeling and simulation makes it feasible to deal with incomplete information. He summarized classic technologies such as fuzzy qualitative simulation, reduction reasoning, noncausal reasoning, causal-based reasoning, diagram-based reasoning, structural data-based modeling, and qualitative space-based reasoning. The application fields were also introduced, including ecology, mechanical manufacturing, medical research, and hybrid nonlinear systems [18]. In [19], a modeling method based on the relationship and transmission of effect between nodes was introduced. Based on the strength of the definition of cause and effect, a flexible modeling method was designed for graph-based qualitative systems. Nonautonomous systems changing with time can be analyzed using the new method. A causal relationship chart model of the quality risk based on integrating casual is proposed in [20]. An example is used to demonstrate the entire risk evolution triggered by changes in one quality factor, simulating the evolution process in accordance with reality. The application indicates that the proposed method can serve as a useful experimental tool for decision making in facing risks by highway construction project teams. A qualitative simulation model of changing processes of customer churn is constructed based on the causality graph in [21]. The qualitative simulation and random behavior extraction can be executed repeatedly to predict the changing process of customer churn. After analyzing three qualitative simulation methods, noncausality reasoning, causality reasoning, and cellular automata, Hu and Xiao discussed the complexity characteristics of a management system and introduced their qualitative simulation [22].

2.2. Qualitative and Quantitative Integrated Modeling in Stochastic Simulation. Many important theories and applications show that qualitative and quantitative combined methodologies have extremely important significance and promote value for stochastic simulation. A considerable amount of researches have been performed in recent years and many meaningful outcomes have been put forward in different domains. In [23], the proposal and recent development of the "meta-synthetic methodology from qualitative to quantitative" were introduced in detail. Subsequently, many researchers concentrated on qualitative and quantitative combined modeling for stochastic simulation.

FAN Shuai proposed a qualitative and quantitative synthetic modeling method by extending the System High Level Modeling Language for multidiscipline virtual prototype. The qualitative knowledge is modeled based on a cause and effect diagram [3]. Then, the qualitative and quantitative integration simulation architecture was designed, including mixed schedule strategies, time management, and date interaction methods [4]. Qualitative models can be built using the Fuzzy Inductive Reasoning paradigm in Modelica. The qualitative models make use of fuzzy inductive reasoning. The qualitative and quantitative models can be combined to simulate concurrently. A textbook example of a hydraulic position control system and the human cardiovascular system were adopted to demonstrate the approach. The hemodynamics was modeled by quantitative models and the central nervous system was described using qualitative FIR models [5]. In [6], a qualitative and quantitative hybrid model was established for business factors evaluation. Statistical values based on propagation and combination of effects of business factors were introduced in the simulation. Li et al. proposed architecture of qualitative and quantitative comprehensive modeling and studied joint simulation technology for complex systems. In [7], a visualized fuzzy qualitative knowledge modeling method fuzzy causal directed graph was designed, which included the grammar, reasoning, and conversion of qualitative and quantitative models. In [8], a new technique, Q2, was proposed to combine qualitative and quantitative models and was demonstrated in the case of a Finnish transport sector that faces severe pressure to cut  $CO_2$  emissions. Liu et al. studied the integration of qualitative reasoning and quantitative simulation including the acquisition, management, and expression of qualitative and quantitative knowledge. Then, an integrated diagnosis inference method was proposed and validated with the test-fire data of complicated systems [9].

Some of the previous studies can be applied to continuous systems, discrete systems, or continuous discrete hybrid system modeling, respectively. They focus on the combination of qualitative and quantitative models from specific application fields. Some researchers achieve qualitative and quantitative combined modeling and simulation based on commercial software tools.

2.3. The COllaborative SIMulation Modeling Theory. COSIM is actually an application of Model Driven Architecture (MDA) for stochastic modeling and simulation. It is mainly a framework for simulation of complex systems, especially

complex product virtual prototypes based on heterogeneous models of different fields. In [16], the modeling specification was proposed as the infrastructure of COSIM which is referred to as the Meta Modeling Framework ( $M^2F$ ). Here, the meta-meta-model, meta-model, and model of different levels were defined to describe the systems. The modeling specification is independent of the realization, which means were that various modeling methods could be involved in the simulation and can be unified with the  $M^2F$  without considering implementation issues. Meanwhile,  $M^2F$  serves as a shield to the differences of the modeling methods and forms with higher abstraction than the heterogeneous model.

2.4. Summary. We can see from the aforementioned that there are many researches on optimization of complex stochastic simulation based on qualitative models. The latest research involves the study of a specific application in a given a field based on a selected theory. Many theories such as reduction reasoning, noncausal reasoning, and causal-based reasoning are considered, respectively. Some researchers achieve qualitative and quantitative combined modeling and simulation based on a commercial software tools. There are mainly two ways to integrate the qualitative and quantitative models, microintegration and macrointegration. The former one extended quantitative description method for qualitative knowledge, usually in the form of qualitative and quantitative mixed algebra equation, such as interval values expression and fuzzy mathematical. Although some qualitative knowledge is used, they are not the systematic qualitative modeling approaches. The later one is the integration of qualitative models and quantitative models of the different parts of the system. For example, qualitative model and quantitative model can be organized together to form the whole simulation system. These methods are mainly integrating different models in particular application, and few of them consider the problem from the aspects of modeling specification. So, a further solution is needed based on the existing theories and techniques.

#### 3. Qualitative and Quantitative Integrated Modeling Specification

3.1. General View of the Qualitative and Quantitative Integrated Model. Before further details, let us first briefly illustrate the general view of the qualitative and quantitative integrated model to be built. We describe a system from the perspectives of static structure and dynamic behavior, based on three types of Interface which is the solid basis of our modeling methodology. The static structure refers to the internal factors, their structure, and interrelationship, for example, the input and output interfaces and their connection relation and the organization structure of the subsystem, and so forth, as shown in Figure 1. The data exchange between the models is archived through the PortItems, and the collection of PortItems with same type is called Port. The set of Ports is called Interface. The three Interfaces (qualitative, quantitative, and event interface, resp.) will have complex internal and external relations with each other, and this is one

FIGURE 1: Modeling the complex system with component-oriented qualitative and quantitative integrated model.

Output function Output

聯

. >

Qualitative PortItem

Adapter



FIGURE 2: A simulation system integrated by qualitative and quantitative models.

of the focus points in this paper. To describe the temporal logic and simulation process with the time advancing, the state and its transfer, interaction situation, and event flow will be modeled as dynamic behavior, and the reasoning or evaluation functionality will also be involved if needed. We can observe a corresponding output segment from the output interface when some data is set from the input interface, taking the data context into account.

A complex system as a whole is composed by many interconnected and interacted parts, and it can be further divided into smaller and simpler subsystems. It is modeled by component-oriented models with a hierarchy structure. Two types of component model with different structure and size are defined to describe the system, named the element model (EM) and composition model (CM). EM is the smallest one which cannot be divided any more, while the CM is assembled by EMs and/or smaller CMs according to specific simulation logic by connecting their Interfaces, and then they can collaborate with each other based on an accurate information flow with specific semantics, as shown in Figure 2. In fact the entire simulation system itself is the biggest CM, with a special reasoning component to optimize the simulation process and policy decision based on execution data, history data, and expertise.



FIGURE 3: The hierarchical model structure of  $Q^2 M^2 F$ .

3.2. Qualitative and Quantitative Integrated Meta Modeling Framework. Based on  $M^2F$ , a Qualitative and Quantitative Integrated Meta Modeling Framework ( $Q^2M^2F$ ), consistent with (MDA) and the rationale of a layered model structure in Meta Object Function (MOF), is defined as a four-layer model framework, as shown in Figure 3. The descriptions of the layers are as follows.

- Meta-Meta Model Layer. The prototypes and rules of the meta-model are defined with the highest abstraction level, including Port, Association, Constrain (CAP). The basic factor and its semantics to describe the data structure and knowledge are also defined, just as the basic data type is defined in a programming language.
- (2) *Meta-Model Layer*. The instance of the meta-meta model, *Mapping*, *Interface*, *Coupling* (CIM), defines the basic factor to define a qualitative and quantitative mixed model. It is similar to defining a data structure or class.
- (3) *Model Layer*. The instance of the meta-model, is used to describe the models (so called *High-Level Model*, HLM) of a specific application field. For example, class "Pilot," a model of the reasoning portion of an expert system, and so forth.
- (4) *Instance*. The instance of the model defines the value of specific parameter or the reasoning part with specific rules, for example, "Pilot Obama."

In  $Q^2M^2F$ , the basic factors in the meta-meta layer are the same as COSIM, but the connotations are redefined to support qualitative and quantitative combined modeling. *Logical Relation* (*LR*) is added to the meta-model layer

Input

**KHAH** 

**Att** 

AXX AXX

Time

ΕM

Time advance

Inner state

State transfer,

reasoning

evaluation

Quantitative PortItem

▹ Event PortItem

to describe the relation of qualitative knowledge. The interaction between qualitative knowledge and quantitative data is added in *Mapping* and *Coupling*. Accordingly, in the model layer instance these factors are also defined.

3.3. The Description Logic System for  $Q^2M^2F$ . Description logic is used to represent the domain knowledge using a group of structural operators. Knowledge is expressed by concepts and relationships based on the formal reasoning which can be achieved [24–26]. In order to describe the basic factors and their relationships in  $Q^2M^2F$ , a description logic system,  $ALC_C$ , is defined based on the classical description logic language, Attributive concept Language with Complements (ALC). The syntactic and semantic facets of  $ALC_C$  are defined as follows:

 $C, D ::= C |\top| \perp |\neg C| C \sqcap D| C \sqcup D| \forall R.C| \exists R.C,$ 

where

*C*, *D*: the elementary concept. In  $Q^2M^2F$ , the term "element" refers to the smallest atomic model,

R: the elementary binary relation,

 $\top$ : the universal concept,

 $\perp$ : the bottom concept,

 $\neg C$ : the negative concept of C,

 $C \sqcap D$ : the intersection of C and D,

 $C \sqcup D$ : the union of C and D,

 $\forall R.C$ : restricted universal quantification,

 $\exists R.C:$  restricted existential quantification.

The knowledge base of  $ALC_C$  is composed of  $\langle T_C, A_C \rangle$ .  $T_C$  is a finite set of inclusion assertion  $(T_{box})$ , and it is also known as a set of terminology axioms.  $A_C$  is a finite set of instance assertion  $(A_{box})$ . It is composed of elementary conception (*ElemC*) and elementary relationship (*ElemR*), as follows:

 $A_C = \langle ElemC, ElemR \rangle.$ 

 $ElemC = \{Data, Knowledge, Event, Input, Output, Time, Real, Pattern, Association, streig (streig, which means "tied or bound" in ancient Latin. Here it is used to represent a constraint.), <math>T_0$ , DataType, KnowledgeType, EventType, STATE, statTF, ID,  $M_x$ }

ElemR = {has\_a, part\_of, domain\_of, range\_of, isa\_function, isa\_relation\_on, content\_of, direction\_of, time\_of, element\_of},

where

*Data, Knowledge* and *Event*: quantitative data, qualitative knowledge and event, respectively;

*Input* and *Output*: the direction of information flow;

*Time*: the effective time of the information flow;

Real is the real numbers;

Pattern: the overall scheme of information;

Association and streig: the Association relationship and constraint, respectively;

 $T_0$ : the initial time;

*STATE* and *statTF*: the state and its transfer, respectively;

*ID*: the index set of the subcomponents;

 $M_X$ : the set of subcomponents;

*has\_a* and *part\_of:* two inverse elementary relationships, expressing the belonging relationship between the elements of the sets;

*domain\_of* and *range\_of*: the domain and range of the relation;

*isa\_function*: a common function;

*isa\_relation\_on*: a binary relation;

*content\_of*: the information of a meta-meta model;

*direction\_of*: the direction of the information;

*element\_of*: the relationship between EM and CM.

More complex conceptions and relationships can be derived from the basic definition mentioned earlier, and the factors at each level in  $Q^2 M^2 F$  can be described and verified formally.

3.4. Meta-Meta Model (CAP). Qualitative and quantitative meta-meta model is the top level of abstraction of the system model. Port is a meta-port composed by Content, Direction, Time, and Pattern and is used to describe the information interaction with other simulation models or the external environment. Content is all the information interacting between the simulation models through the Port which will affect the simulation process or result. Content can be quantitative data, event, or qualitative knowledge. Direction indicates the transfer direction of the information. Time refers to the position and effective range on the timeline. The value range T is a subset of the positive real numbers  $R^+$ . Pattern describes the overall pattern of information contained by meta-ports throughout the simulation timeline. It is an enumerable sequence of a set of numerable/innumerable (content, time) couples. The formal definition of Port is as follows:

Port  $\equiv \exists has\_a.Content \sqcap \exists has\_a.Direction \sqcap \exists has\_a.Time \sqcap \exists has\_a.Pattern$ Content  $\equiv Data \sqcup Knowledge \sqcup Event$ Direction  $\equiv Input \sqcup Output$ Time  $\sqsubseteq Real$ Pattern  $\sqsubseteq Content \times Time.$ (Meta) Association is used to describe the numeri-

(*Meta)* Association is used to describe the numerical/symbolic relationship of information contents between *meta-ports*. The association represents the direction of the *Content*, and most of the association is a one-to-one mapping. In quantitative models, the association is reflected as a mapping relationship between quantitative data on the metaports. In qualitative models, it is the *connecting* relationship between qualitative knowledge. Multiple associated *ports* may also exist, which represent the convergence or distribution of the information flow. The formal definition is

Association  $\sqsubseteq$  Port\_Content × Port\_Content Port\_Content  $\equiv$  Content  $\sqcap \exists part_of.Port.$ 

*Constrain* describes the properties of specific *Port*, including differences in *Direction*, *Time*, and *Pattern*, especially for the ports where *Association* exists. There are two *Constraints*, *Quantitative Constraint* and *Qualitative Constraint*. By setting constraints on the *Direction*, *Time*, and *Pattern*, the solution logic, temporal order, and modeling mechanism of heterogeneous models can be unified in one simulation system. *Constraint* will be implemented according to the interior physical mechanism or the state transfer function in the lower layer *HLM*. The conception of *Constrain* is defined as

 $Port\_Direction \equiv Direction \sqcap \forall part\_of.Port$  $Port\_Pattern \equiv Pattern \sqcap \forall part\_of.Port$  $Port\_Time \equiv Time \sqcap \forall part\_of.Port$  $Constrain \equiv streig \sqcap (\exists isa\_function.Port\_Direction \sqcup \exists isa\_function.Port\_Pattern \sqcup \exists isa\_function.Port\_Time).$ 

In summary, the concept of CAP is defined formally as

 $CAP \equiv \exists has\_a.Port \sqcap \exists has\_a.Association \sqcap \exists has\_a.Constrain.$ 

3.5. Meta-Model (CIM). We define PortItem, Ports, and Interface as instances of Port in the CIM model. PortItem is consistent with Port, Ports are defined as a collection of PortItems of the same type, and the Interface is defined as a group of Ports with similar properties. This is formally defined as

 $PortItems \equiv \exists part\_of.CAP \sqcap Port$ 

 $PortItem \equiv \exists part\_of.PortItems$ 

Association and Constrain are essentially interdependent of each other. The former characterizes the existence of the information relationship between the *Ports*, while the latter adds a limitation on the relationship. There are three instances, *Mapping*, *Logical Relation*, and *Coupling*, in the meta-model inherited from both *Association* and *Constrain*.

*Mapping*, a coinstance of *Association* and *Constrain*, is a relationship between the input and output sets of an element model (*EM*). Figure 4 shows three typical *Mappings*, the state transfer functions between quantitative *PortItems* (*map*), logical relationship between qualitative *PortItems* (*connect*), and *transform* between quantitative and qualitative *PortItems*. The definition is as follows:



FIGURE 4: Three typical Mappings in an EM.

 $Mapping \equiv Maps \sqcup Connects \sqcup Transforms$ 

 $Connects \equiv \exists part_of.CAP \sqcap Association \sqcap \exists isa_relation_on.(\exists content_of.Knowledge)$ 

The relationship between the qualitative *Ports* is not necessarily expressed via functions; general logical relationships may exist. *Logical Relations* mainly depicts the qualitative relationship between *Ports* and the static logical structure of an *EM*. Consider

 $LogRelation \equiv (connect | connect \in {Interface_i. Content} \times {Interface_j. Content}, i \neq j, Interface_j, Interface_i \in EM \cup CM \rangle.$ 

*Coupling* is the interaction between the *Ports* containing the *Associations* and *Constraints*. In addition, it should be noted that the ports associated by *Coupling* are not just the ports of the submodels within a composite model. *Associations* could also exist between the output ports of the submodels and the output ports of its superior composite model. Similarly, the input ports of a composite model can be associated with the input of its submodel. The formal definition of *Coupling* is

 $Coupling \equiv Coupling\_maps \sqcup Coupling\_connects$ 

In summary, the concept of CIM is defined formally as

 $CIM \equiv \exists has\_a.Interface \sqcap \exists has\_a.Mapping \sqcap \exists has\_a.Coupling.$ 

3.6. The Hierarchy Model of a Simulation System (HLM). A variety of heterogeneous simulation functionalities are described as standard models using an interface-based modeling strategy. Simulation is achieved via the combination and collaboration of components. In the model layer, the simulation model, named high level model (HLM), will be instanced from three basic factors defined at the meta-model layer. There are two types of qualitative and quantitative mixed simulation models, the Element Model (*EM*) and the Composite Model (*CM*).

As the smallest model which cannot be divided any more, *EM*, consists of *Interface*, *Mapping*, *and Connecting*, the definition is:

EM :  $\langle \{Interface\}, \{Mappings, Connectings\} \rangle$ .

More specifically,

$$\begin{split} EM &\equiv \exists has\_a.(Init) \sqcap \exists has\_a.(iP_d) \sqcap \exists has\_a.(iP_k) \sqcap \\ \exists has\_a.(iP_e) \sqcap \exists has\_a.(oP_d) \sqcap \exists has\_a.(oP_k) \sqcap \exists has\_a.(oP_e) \sqcap \exists has\_a.(STATE) \sqcap \exists has\_a.(statTF) \sqcap \exists has\_a.(T), \end{split}$$

where

 $Init \equiv \exists part_of.CIM \sqcap In\_PortItem \sqcap \exists has\_a.(\exists time\_of.T_0) \sqcap \exists has\_a.DataType$ 

 $iP_d \equiv \exists part_of.CIM \sqcap In\_PortItem \sqcap \exists has\_a.(\exists content_of.Data) \sqcap \exists has\_a.DataType$ 

 $iP_k \equiv \exists part_of.CIM \sqcap In\_PortItem \sqcap \exists has\_a.(\exists content_of.Knowledge) \sqcap \exists has\_a.KnowledgeType$ 

 $iP_e \equiv \exists part_of.CIM \sqcap In\_PortItem \sqcap \exists has\_a.(\exists content_of.Event) \sqcap \exists has\_a.EventType$ 

 $oP_d \equiv \exists part\_of.CIM \sqcap Out\_PortItem \sqcap \exists has\_a.(\exists content\_of.Data) \sqcap \exists has\_a.DataType$ 

 $oP_k \equiv \exists part_of.CIM \sqcap Out\_PortItem \sqcap \exists has\_a.(\exists content_of.Knowledge) \sqcap \exists has\_a.KnowledgeType$ 

 $oP_e \equiv \exists part\_of.CIM \sqcap Out\_PortItem \sqcap \exists has\_a.(\exists content\_of.Event) \sqcap \exists has\_a.EventType.$ 

STATE represents a specific mapping between the input and output *Ports*. At any time t on the timeline T, the simulation model has only one state, *csModelState* (t), and the formal definition is

 $STATE \equiv \exists csModelState. T,$ 

where

*StatTF*: state transfer refers to the migration process stimulated by external action or internal factors;



FIGURE 5: The structure of a CM.

5

$$StatTF = \langle STATE \times iPe \rightarrow STATE \times T \times oPe \mid T \subseteq \mathbb{R}^+ \rangle.$$

A *Composite Model* (*CM*) is composed of several *EMs* and/or *CMs* with smaller granularity as shown in Figure 5. The formal definition is as follows:

 $\begin{array}{l} CM \equiv \exists has\_a.(Para) \sqcap \exists has\_a.(Init) \sqcap \exists has\_a.(iP_d) \sqcap \\ \exists has\_a.(iP_k) \sqcap \exists has\_a.(iP_e) \sqcap \exists has\_a.(oP_d) \sqcap \exists has\_a.(oP_k) \sqcap \exists has\_a.(oP_e) \sqcap \exists has\_a.(T) \sqcap \exists has\_a.(ID) \sqcap \exists has\_a.(M_x) \sqcap \exists has\_a.(CPLs) \sqcap \exists has\_a.(SITUA) \sqcap \exists has\_a.(EvntFL). \end{array}$ 

Similarly with *EM*, *CM* also has a parametric interface, initialization interface, data input and output interfaces, event input and output interfaces, knowledge input and output interfaces, the state and its transfer, and the time-base. They are defined as earlier. *CM* has three other factors, *ID*,  $M_X$ , *SITUA*, *EvntFL*, and *CPL*<sub>S</sub>, which do not appear in *EM*, as follows:

*ID*: the index set of the sub-*EM*/sub-*CM* in a *CM*,

 $M_X$ : the set of sub-*EMs* and sub-*CMs* in a *CM*,

CPLs: the Coupling sets in a CM,

SITUA: the sets of interaction situation,

*EvntFL*: event flow in a *CM*.

We can see that CM is a self-nested composite model. Besides the element model, EM, which can no longer be divided, it can also include other composition models. In fact, the whole simulation system itself is the largest CM.

#### 4. The Self-Closed Feature of Qualitative and Quantitative Integrated Model

We can find that the essential difference between the EM and CM is whether or not an internal structure exists. EM describes the internal content of a model via mappings, while CM describes its interior structure and interactions among

subcomponents. Formally, there are few differences between the two models, but we can note that the formalism of a *CM* actually has a self-closed structure. Although the internal structure of a *CM* might be very complicated, a *CM* should be reused just like an *EM* in a more complex *CM*. Therefore, in order to ensure reusability, we need to affirm the self-closed feature between the *EM* and *CM*. That is, a complicated *CM* combined by sub-*EM* and/or sub-*CM* has the same schema as its subcomponents. On the contrary, the subcomponents decomposed from a *CM* has the same schema with the original *CM*.

*Definition 1* (Component Communication Graph (*CCG*)). Assume *C* is a simulation component (*CM* or *EM*). Let directed graph  $G_C = \langle V_C, E_C \rangle$  be the *CCG* of *C*. Consider

$$V_C = input\_interface(C) \cup output\_interface(C),$$

 $input\_interface(C) = \{x \mid (EM(C) \land input(x, e)) \lor \exists e(EM(e) \land element\_of(e, C)) \land input(x, e))\},\$ 

*input\_interface* (*C*) is the set of all input interfaces of *C*. If *C* itself is a *CM*, all input interfaces of the internal subcomponents are the same as well,

 $output\_interface (C) = \{x \mid (EM(C) \land output (x, e)) \lor \exists e(EM(e) \land element\_of (e, C) \land output (x, e))\}.$ 

The edge set  $E_C$  is

 $E_{C} = \{ \langle x, y \rangle \mid (EM(C) \land \langle x, y \rangle \in Mapping_{C}) \lor (CM(C) \land \langle x, y \rangle \in Coupling_{C}) \lor \exists e(EM(e) \land element\_of (e, C) \land \langle x, y \rangle \in Mapping_{e}) \}.$ 

The previous definition shows that the vertex set  $(V_C)$  of *CCG* is composed of all input and output *Interfaces* of the high level model, and  $E_C$  is composed of all the *Mappings edges* and *Coupling edges*. If there is a *Mapping* or *Coupling* between two Interfaces, the two vertices are adjacent.

*Definition 2 (Maps*  $_C$  and *Couples*  $_C$  of *CCG*). Consider the following:

if component C is an EM,  $Maps_C = Mapping_C$ ,  $Couples_C = \emptyset$ ; if component C is a CM,  $Maps_C = \sum_{e \in C} Mapping_e$ ,  $Couples_C = Coupling_C$ .

*Deduction 1.* The underlying graph of  $G_C = \langle V_C, E_C \rangle$  is a bipartite graph.

Ignoring the direction of all the edges of the directed *CCG*, we can get its underlying graph. We can prove that the underlying graph of *CCG* is a bipartite graph.

Let *X* = input\_interface(*C*), *Y* = output\_interface(*C*),

 $\Rightarrow V = X \cup Y \text{ and } X \cap Y = \emptyset,$ 

=> X and Y is 2-partition of  $V_{\rm C}$ .

According to Definition 1,

 $\begin{array}{ll} \forall x \forall y (xy \in E(G_C) \rightarrow (\exists e(EM(e) \land \langle x, y \rangle \in Mapping_e) \lor (CM(C) \land \langle x, y \rangle \in Coupling_C)). \end{array}$ 

According to Definition 2,

$$\Rightarrow \forall x \forall y (xy \in E(G_C) \rightarrow ((x \in X \land y \in Y) \lor (x \in Y \land y \in X)),$$

 $\Rightarrow G_C = \langle V_C, E_C \rangle$  is a bipartite graph.

The previous deduction means that the *Mapping* connects the input and output *Interfaces* of an *EM*, and *Coupling* connects the input and output *Interfaces* between *EMs* and/or *CMs*. The vertices of *X* are independent of each other, and vertices of Yare also independent.

*Definition 3* (Information Tracking). Let  $G_C = \langle V_C, E_C \rangle$  be the CCG of C,  $x_0 \in V(G_C)$ ,  $x_k \in V(G_C)$ , if

 $P = x_0 m_1 x_1 m_2 x_2 \cdots m_k x_k \land \forall i = 1, 2, \dots, k (m_i = \langle x_{i-1}, x_i \rangle) \land \forall i \forall j (i \neq j \rightarrow x_i \neq x_j).$ 

Then *P* is Information Tracking in  $G_C$ .  $x_0$  and  $x_k$  are the start and end points of *P*, referred to as *startPoint*<sub>*P*</sub> and *endPoint*<sub>*P*</sub> respectively.

Using the terminology of graph theory, *Information Tracking* can be described as follows:

Vertex *i* and *j* ( $i \neq j$ ) belong to  $G_C$ , and *P* is a primary path from *i* to *j* without repetitive vertices. If any adjacent vertex of *x* is from the same *EM* with *x*, then it is an *Information Tracking* of  $G_C$ .

When there are only two vertices in the  $Maps_C$  or  $Couples_C$ , we can easily get the following deduction.

*Deduction 2. Mapping* and *Logical Relation* are both a kind of *Information Tracking*.

We can see from Definition 3 that *Information Tracking* is a directed path, the direction of  $Maps_C$  is always from the input *Interface* to the output *Interface*, while the direction of *Couples*<sub>C</sub> is from output to input. In the *Information Tracking P*, the edges of  $Maps_C$  and  $Couples_C$  appear alternately.

Deduction 3. Information Tracking  $P \in InfoPath_C$  and  $P = x_0 m_1 x_1 m_2 x_2 \cdots m_k x_k$ ,  $i \in \{1, 2, \dots, k\}$ ; if  $m_1 \in Maps_C$  then  $m_i \in Maps_C$  if and only if  $i \equiv 1 \pmod{2}$  and  $m_i \in Couples_C$  if and only if  $i \equiv 0 \pmod{2}$ ; if  $m_1 \in Couples_C$  then  $m_i \in Couples_C$  if and only if  $i \equiv 1 \pmod{2}$  and  $m_i \in Maps_C$  if and only if  $i \equiv 1 \pmod{2}$  and  $m_i \in Maps_C$  if and only if  $i \equiv 0 \pmod{2}$ .

Deduction 1=> the underlying graph of  $G_C$  is a bipartite graph,

Deduction  $2 \Rightarrow$  in *Information Tracking P*, the edges of *Maps*<sub>C</sub> and *Couples*<sub>C</sub> appear alternately.

Assume that *xy* and *yz* are two adjacent edges of *P*, a primary path. So,  $x \neq z$ .

If  $xy \in Maps_C$ , then x and y are the input and output of an EM(c). The vertices x and z are adjacent to y.

=> In *x* and *z*, one must belong to EM(c), and  $x \neq z$ .

=> *z*must not be the Interface of *EM*(*c*), and it must belong to other *EM*/*CM*.



FIGURE 6: An *Information Tracking* composed by alternative *Mapping* and *Coupling*.

The underlying graph of  $G_C$  is a bipartite graph, and in *Information Tracking P*, the edges of  $Maps_C$  and  $Couples_C$  appear alternately.

- => z is a input *Interface*.
- $\Rightarrow$  yz  $\in$  Couples<sub>C</sub>.

If  $xy \in Couples_C$ , then x and y are the input and output of two different *components*. Assume that y belongs to EM(c1).

- => In x and z, there must be one belonging to EM(c1), and  $x \neq z$ .
- => z must not be the Interface of EM(c1), and it belongs to the other EM.

The underlying graph of  $G_C$  is a bipartite graph, and, in *Information Tracking P*, the edges of  $Maps_C$  and  $Couples_C$  appear alternately.

- => *z* is a output *Interface*.
- $\Rightarrow$  yz is a *Mapping* of the other *EM*, yz  $\in$  *Maps*<sub>C</sub>.
- => P is an uninterrupted path composed of alternative Mapping and Coupling. It can also be expressed by alternative input and output Interface, as shown in Figure 6.

The vertices in *P* are independent of each other and  $i \neq j$ , and  $\forall_i \forall_j (i \neq j \rightarrow x_i \neq x_j)$ .

- => *P* is a directed path without repetitive edges and there is no closed loop in *P*.
- => If  $m_1 \in Maps_C$ ,  $m_i \in Maps_C$  if and only if  $i \equiv 1 \pmod{2}$  and  $m_i \in Couples_C$  if and only if  $i \equiv 0 \pmod{2}$ , and if  $m_1 \in Couples_C$ ,  $m_i \in Couples_C$  if and only if  $i \equiv 1 \pmod{2}$  and  $m_i \in Maps_C$  if and only if  $i \equiv 0 \pmod{2}$ .

Definition 4 (Derivable Port and Underivable Port).  $G_C = \langle V_C, E_C \rangle$  is the CCG of C,  $x \in$  output\_interface<sub>C</sub>. If  $\exists P(P \in InfoPath_C \land P = x_0 m_1 x_1 m_2 x_2 \cdots m_k x \land x \in input_interface_C)$ ), then x is a Derivable Port of C, or x is an Underivable Port (referred to as DerivablePorts (C) and UnderivablePorts (C), resp.).

Both Derivable Ports and Underivable Ports are output Ports. For output Ports, there are input Ports connected to it



FIGURE 7: Scenario of the stochastic defense simulation system.

through an *Information Tracking*, but there is no such input for an *Underivable Port*. The internal mechanism and status of a black-box model are normally undetectable. Some outputs might be generated without any inputs and the only reason for this is due to the internal state transfer driven by time. That is why an underivable *Port* is needed.

Deduction 4.  $G_C = \langle V_C, E_C \rangle$  is the CCG of C,  $x \in DerivablePorts(C)$ .  $\exists e \ (e \in EM \land element_of \ (e,C) \land x \in UnderivablePorts \ (C))$  or  $\exists e \ (e \in EM \land element_of(e,C) \land \exists x_0 \exists P(P \in InfoPath(C) \land x_0 \in UnderivablePorts \ (C) \land x_0 = startPoint_P \land x = endpoint_P)).$ 

In a white-box CM(C), let j be the Underivable Port of  $G_C$ . Every output Port of CM(C) is connected with an output Port of an internal EM(C) by Coupling. Assume that Port j of CM(C) is connected with output Port i of EM(C), as shown in Figure 4. We will prove that Port i is an Underivable Port by reducing it to absurdity.

Assume that *Port i* is a *derivable Port*, and then there is an *Information Tracking* in  $G_C$ . *Port i* is the endpoint of *P*.

- *Port i* is the output *Port* of EM(C).
- => The last edge of *P* must belong *to Maps<sub>C</sub>* (Deduction 3).
- =>  $P' = P \cup \{ij\}$  is another *Information Tracking* in  $G_C$  and at least one input *Port* of P' comes from  $G_C$ .
- => The end point *j* of *P*′ is a derivable *Port*. This is contradictory.
- => Port i is an Underivable Port.
- => Underivable Port exists in an EM, and a Port connected to an Underivable Port by Coupling is also an Underivable Port.

In summary, we can see that



FIGURE 8: Qualitative and quantitative combined models of the stochastic defense simulation system.



FIGURE 9: The reasoning *EM*, extended Fuzzy CLIPS as the reasoning engine.

$$\begin{split} EM &\sqcap interface \equiv \exists has\_a.(Init) \sqcap \exists has\_a.(iP_d) \sqcap \exists has\_a.(iP_k) \sqcap \exists has\_a.(iP_e) \sqcap \exists has\_a.(oP_d) \sqcap \exists has\_a.(oP_k) \sqcap \exists has\_a.(oP_e) \end{split}$$

$$\begin{split} CM &\sqcap interface \equiv \exists has\_a.(Init) \sqcap \exists has\_a.(iP_d) \sqcap \exists has\_a.(iP_k) \sqcap \exists has\_a.(iP_e) \sqcap \exists has\_a.(oP_d) \sqcap \exists has\_a.(oP_k) \sqcap \exists has\_a.(oP_e) \end{split}$$



FIGURE 10: Survival probability of target.

 $=> EM \sqcap interface \equiv CM \sqcap interface.$ 

This means that *EM* and *CM* have the same schema, and the *HLM* is self-closed.

#### 5. A Stochastic Defense Simulation System

5.1. The Scenario and Integrated Models. In the simulation, the attacker and the target patrol in the same area, and both of them have detecting ability. As the distance between the two sides becomes shorter, the attacker will find the target and launch its offense weapon which will seek the target using its detector. The offense weapon will rush out with full speed when it detects the target. After detecting the offense weapon, the target will launch decoy or evade with different direction and speed according to the defense strategy. The scenario is as follows (Figure 7). Many previous researches carried out the same scenario; however, most of them focused on using fuzzy logic to make the decision [1] in a specific application or evading in a fixed manner [27].

The system is modeled using the proposed specification. We designed nine *EMs* as shown in Figure 8. There are three *CMs*, *Attacker*, *Target*, and *Offense Weapon*, which are composed by two *EM* models, respectively. The data, event, and knowledge interactions among the *CMs/EMs* are also given in the figure. Different shapes are used to describe different types of *Ports*. The circle, square, triangle, and oval *Ports* represent initialize port, event port, data port, and knowledge port, respectively. What should be pointed out is that only *Ports* and *Couplings* are given in the figure, not the *PortItems* and *Mapping*. *Mappings* are inside the *EM* and are invisible from the outside. Due to the space, the dynamic behavior and schedule of *CM/EM* will be treated as a blackbox and will be discussed in the future.

5.2. Optimization of the Defense Simulation. In our simulation, defense strategy and simulation operation strategy are decided by reasoning EM based on the real-time battlefield situation and expert experience to optimize the simulation result. We have several evasion strategies, such as launching a decoy at specific position with a reasonable direction and moving mutely with higher speed alone against\_direction, depending on the battlefield situation, the decoy status, and distance between the offence weapon and target. Different simulation strategies could be adopted in different situations to optimize the operating efficiency. When the attacker is far away from the target and any other special task, the simulation can run with super-real-time speed (in speedup status); only some staple detectors in work and many functionalities will not be executed or executed in less time (in light\_caculate status). The simulation time will slow down when the attacker gets closer to the target and more powerful detector will be on duty.

The defense strategy and simulation operation strategy are decided by a reasoning *EM*. The detail is as follows (Figure 9). We proposed a new fuzzy-reasoning algorithm based on confidence fuzzy rules and embedded it into Fuzzy CLIPS. The extended Fuzzy CLIPS is encapsulated into the *EM* as a reasoning engine. The rules coming from expert knowledge are stored as a file (\*.clp) and will be loaded to the rule base. At running time, different strategies will be made according to the battlefield situation. Some of the confidence fuzzy rules are as follows. *Rule 1. IF* Weapon\_distance\_medium *AND* Decoy1\_ready *THEN* Change\_Direction\_with\_large\_angle AND evade\_ mutely *AND* Launch\_Decoy1, *Confidence:* 0.85.

*Rule 2. IF* Weapon\_distance\_short *THEN* Evade\_full\_speed, *Confidence:* 0.9.

*Rule 3. IF* Distance\_between\_attacker\_target\_far *THEN* simulation\_speedup AND light\_caculate, *Confidence:* 0.9.

÷

5.3. Simulation Results and Analysis. The initial speeds of attacker and target are both 18 m/s. When the offence weapon is launched, its initial speed is 20 m/s. The detection range is 1.5 km apart. The initial distance between attacker and target is 8 Km. The simulation is executed in two situations. First, defense strategy is fixed as evade\_full\_speed or Lauch\_Decoy1 or Lauch\_Decoy2 randomly and running speed is also fixed. Secondly, the reasoning model will be used. The simulation time and data communication can be saved significantly at the beginning because of simulation\_speedup and light\_caculate strategy.

In fact, the voyage of the weapon is one of the key factors in the survival probability of the target. If the voyage is long enough, the target will be destroyed with probability 1. If it is short, the weapon will exhaust before catching the target. We set different voyages for the weapon, and the simulation is executed 20 times for each voyage in the two situations. The average survival probability is shown in Figure 10. We can see that when the voyage is shorter than 8140 m, the target will always survive, and if the voyage is longer than 8380 m, the target will be destroyed absolutely. Between 8380 m and 8140 m, the probability of survival is higher, when we simulate based on qualitative and quantitative integrated models.

#### 6. Conclusions and Future Works

In this paper, we have proposed a new specification to modeling qualitative and quantitative hybrid system for stochastic simulation and optimization. The new specification is defined at three levels and its self-closed feature is proven to be self-closed formally. The definition of factors needed to describe the integrated models and corresponding *Mapping* and *Coupling* is presented in detail. This provides a new way to take advantage of qualitative models in stochastic simulation. A stochastic simulation defense system was modeled and realized using the proposed specification; a reasoning engine is encapsulated as a qualitative *EM* and interacts with quantitative models at running time. The result shows that the hybrid models can optimize the stochastic simulation significantly on both the execution process and the performance.

As future works, the dynamic behavior and schedule engine of qualitative and quantitative integrated models for stochastic simulation in different application should be a great work that will be promoted in detail and verified. Also, more working on the integration relationship, interaction, and time management of qualitative and quantitative stochastic models are significant for the new specification.

#### Acknowledgments

This work was supported by Major Basis Research under Grant no. C0420110005 in China. The authors acknowledge and appreciate all the team members. They are also grateful to editors and reviewers for their constructive comments, which helped improve this paper greatly.

#### References

- M.-J. Son and T.-W. Kim, "Torpedo evasion simulation of underwater vehicle using fuzzy-logic-based tactical decision making in script tactics manager," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7995–8012, 2012.
- [2] L. I. Wenwei, "Research on the development and application of the method of qualitative simulation," *System Simulation Technology*, vol. 4, no. 2, pp. 71–74, 2008 (Chinese).
- [3] S. Fan, B.-H. Li, X.-D. Chai, B.-C. Hou, and T. Li, "Studies on complex system qualitative and quantitative synthetically modeling technologies," *Journal of System Simulation*, vol. 23, no. 10, pp. 2227–2233, 2011.
- [4] S. Fan, B.-H. Li, X.-D. Chai, and X.-D. Huang, "Studies on qualitative and quantitative integration model computing technology," *Journal of System Simulation*, vol. 23, no. 9, pp. 1980–1984, 2011.
- [5] F. E. Cellier, "Mixed quantitative and qualitative simulation in modelica," in *Proceedings of the 7th Modelica Conference*, pp. 86– 95, Como, Italy, September 2009.
- [6] M. Samejima, M. Akiyoshi, K. Mitsukuni, and N. Komoda, "Business scenario evaluation using monte carlo simulation on qualitative and quantitative hybrid model," *Electrical Engineering in Japan*, vol. 170, no. 3, pp. 9–18, 2010.
- [7] T. Li, B.-H. Li, X.-D. Chai, and S. Fan, "Research on knowledge modeling and joint simulation method of complex qualitative system," *Journal of System Simulation*, vol. 23, no. 6, pp. 1256– 1260, 2011.
- [8] V. Varho and P. Tapio, "Combining the qualitative and quantitative with the Q<sub>2</sub> scenario technique—the case of transport and climate," *Technological Forecasting and Social Change*, vol. 80, no. 4, pp. 611–630, 2013.
- [9] H.-g. Liu, J.-j. Wu, and Q.-z. Chen, "Fault diagnosis reasoning based on integration of qualitative reasoning and quantitative simulation," *Journal of System Simulation*, vol. 15, no. 5, pp. 689– 692, 2003.
- [10] A.-F. Shi, Q.-Z. Wu, K. Huang, and H. Zheng, "Study on characteristic of qualitative models and simulation about complex weapon system," *Journal of System Simulation*, vol. 18, no. 2, pp. 591–593, 2006 (Chinese).
- [11] K. Ogata, *Modern Control Engineering*, Prentice Hall, Upper Saddle River, NJ, USA, 5th edition, 2009.
- [12] K. Edström, "Simulation of Newton's pendulum using switched bond graphs," 1998, http://citeseerx.ist.psu.edu/.
- [13] R. David and H. Alla, "On hybrid Petri nets," Discrete Event Dynamic Systems, vol. 11, no. 1-2, pp. 9–40, 2001.
- [14] D. Karnopp, R. Rosenberg, Perelson, and S. Alan, System Dynamics: A Unified Approach, John Wiley & Sons, New York, NY, USA, 1976.

- [15] B. P. Zeigler, H. Praehofer, and T. G. Kim, *Theory of Modeling and Simulation*, Academic Press, New York, NY, USA, 2nd edition, 2000.
- [16] B. H. Li, X. Chai, X. Yan, and B. Hou, Multi-Disciplinary Virtual Prototype Modeling and Simulation Theory and Application, Nova Science, New York, NY, USA, 2012.
- [17] L. Ironi and L. Panzeri, "Qualitative simulation of nonlinear dynamical models of gene-regulatory networks.," in *Proceedings* of the 22nd International Workshop on Qualitative Reasoning, University of Colorado, Boulder, Colo, USA, 2008.
- [18] C.-X. Shao and F.-Z. Bai, "Technology of qualitative simulation and its application," *Journal of System Simulation*, vol. 16, no. 2, pp. 202–208, 2004 (Chinese).
- [19] Y. Hashiura, T. Fujimoto, and T. Matsuo, "Qualitative simulation models for non-autonomous systems," in *Proceedings of the 21st International Conference on Systems Engineering (ICSEng '11)*, pp. 392–397, August 2011.
- [20] H. Yang, B. Hu, and L. Zhang, "Analysis for the quality risk evolution of highway engineering construction based on qualitative simulation," in *Proceedings of the International Conference* on Computer Science and Service System (CSSS '11), pp. 212–215, June 2011.
- [21] H. Wang and W.-J. Li, "Study on the Qualitative simulationbased customer churn prediction," in *Proceedings of the International Symposium on Information Engineering and Electronic Commerce (IEEC '09)*, pp. 528–532, May 2009.
- [22] B. Hu and R. Xiao, "Qualitative simulation for complex systems," *System Simulation Technology*, vol. 2, no. 1, pp. 1–10, 2006 (Chinese).
- [23] R.-w. Dai, "The proposal and recent development of metasynthetic method from qualitative to quantitative," *Chinese Journal of Nature*, vol. 31, no. 6, pp. 311–314, 2009 (Chinese).
- [24] F. Baader, D. L. McGuinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press, Cambridge, Mass, USA, 2003.
- [25] F. Baader, C. Lutz, M. Miličić, U. Sattler, and F. Wolter, "Integrating description logics and action formalisms: First results," in *Proceedings of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference (AAAI '05/IAAI '05)*, pp. 572–577, July 2005.
- [26] A. Borgida, "On the relative expressiveness of description logics and predicate logics," *Artificial Intelligence*, vol. 82, no. 1-2, pp. 353–367, 1996.
- [27] D.-Y. Cho, M.-J. Son, J.-H. Kang et al., "Analysis of a submarine's evasive capability against an antisubmarine warfare torpedo using devs modeling and simulation," in *Proceedings of the Spring Simulation Multiconference and DEVS Integrative Modeling and Simulation Symposium (DEVS)*, pp. 307–315, Norfolk Marriott Waterside, Norfolk, Virginia, USA, 2007.

## Research Article

# Univex Interval-Valued Mapping with Differentiability and Its Application in Nonlinear Programming

## Lifeng Li,<sup>1,2</sup> Sanyang Liu,<sup>1</sup> and Jianke Zhang<sup>2</sup>

<sup>1</sup> Department of Mathematics, School of Science, Xidian University, Xi'an 710126, China
 <sup>2</sup> Department of Mathematics, School of Science, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Correspondence should be addressed to Sanyang Liu; liusanyang@126.com

Received 25 November 2012; Accepted 26 May 2013

Academic Editor: Lotfollah Najjar

Copyright © 2013 Lifeng Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Interval-valued univex functions are introduced for differentiable programming problems. Optimality and duality results are derived for a class of generalized convex optimization problems with interval-valued univex functions.

#### 1. Introduction

Imposing the uncertainty upon the optimization problems is an interesting research topic. The uncertainty may be interpreted as randomness, fuzziness, or interval-valued fuzziness. The randomness occurring in the optimization problems is categorized as the stochastic optimization problems, and the imprecision (fuzziness) occurring in the optimization problems is categorized as the fuzzy optimization problems. In order to perfectly match the real situations, intervalvalued optimization problems may provide an alternative choice for considering the uncertainty into the optimization problems. That is to say, the coefficients in the interval-valued optimization problems are assumed as closed intervals. Many approaches for interval-valued optimization problems have been explored in considerable details; see, for example, [1-3]. Recently, Wu has extended the concept of convexity for real-valued functions to LU-convexity for interval-valued functions, then he has established the Karush-Tucker conditions [4-6] for an optimization problem with intervalvalued objective functions under the assumption of LUconvexity. Similar to the concept of nondominated solution in vector optimization problems, Wu has proposed a solution concept in optimization problems with interval-valued objective functions based on a partial ordering on the set of all closed intervals, then the interval-valued Wolfe duality theory [7] and Lagrangian duality theory [8] for intervalvalued optimization problems have been proposed. Recently, Wu [9] has studied the duality theory for interval-valued linear programming problems.

In 1981, Hanson [10] introduced the concept of invexity and established Karush-Tucker type sufficient optimality conditions for a nonlinear programming problem. In [11], Kaul et al. considered a differentiable multiobjective programming problem involving generalized type I functions. They investigated Karush-Tucker type necessary and sufficient conditions and obtained duality results under generalized type I functions. The class of B-vex functions has been introduced by Bector and Singh [12] as a generalization of convex functions, and duality results are established for vector valued B-invex programming in [13]. Bector et al. [14] introduced the concept of univex functions as a generalization of B-vex functions introduced by Bector et al. [15]. Combining the concepts of type I and univex functions, Rueda et al. [16] gave optimality conditions and duality results for several mathematical programming problems. Aghezzaf and Hachimi [17] introduced classes of generalized type I functions for a differentiable multiobjective programming problem and derived some Mond-Weir type duality results under the above generalized type I assumptions. Gulati et al. [18] introduced the concept of  $(F, \alpha, \rho, d)$ -V-type I functions and also studied sufficiency optimality conditions and duality multiobjective programming problems.

This paper aims at extending the Karush-Tucker optimality conditions to nonconvex optimization problem with interval-valued functions. First, we extend the concept of univexity for a real-valued function to an interval-valued function and present the concept of interval-valued univex functions. Then, the Karush-Tucker optimality conditions are proposed for an interval-valued function under the assumption of interval-valued univexity.

#### 2. Preliminaries

Let one denotes by  $\mathscr{I}$  the class of all closed intervals in R.  $A = [a^L, a^U] \in \mathscr{I}$  denotes a closed interval, where  $a^L$  and  $a^U$  mean the lower and upper bounds of A, respectively. For every  $a \in R$ , we denote a = [a, a].

*Definition 1.* Let  $A = [a^L, a^U]$  and  $B = [b^L, b^U]$  be in  $\mathcal{I}$ ; one has

(i) 
$$A + B = \{a + b : a \in A \text{ and } b \in B\} = [a^L + b^L, a^U + b^U];$$

(ii) 
$$-A = \{-a : a \in A\} = [-a^U, -a^L];$$

(iii)  $A \times B = \{ab : a \in A \text{ and } b \in B\} = [\min_{ab}, \max_{ab}],$ where  $\min_{ab} = \min\{a^L b^L, a^L b^U, a^U b^L, a^U b^U\}$  and  $\max_{ab} = \max\{a^L b^L, a^L b^U, a^U b^L, a^U b^U\}.$ 

Then, we can see that

$$A - B = A + (-B) = \begin{bmatrix} a^{L} - b^{U}, a^{U} - b^{L} \end{bmatrix},$$
$$kA = \{ka : a \in A\} = \begin{cases} \begin{bmatrix} ka^{L}, ka^{U} \end{bmatrix} & \text{if } k \ge 0, \qquad (1) \\ \begin{bmatrix} ka^{U}, ka^{L} \end{bmatrix} & \text{if } k < 0, \end{cases}$$

where *k* is a real number.

By using *Hausdorff metric*, Neumaier [19] has proposed *Hausdorff metric* between the two closed intervals *A* and *B* as follows:

$$d_H(A, B) = \max\{|a^L - b^L|, |a^U - b^U|\}.$$
 (2)

Definition 2. Let  $A = [a^L, a^U]$  and  $B = [b^L, b^U]$  be two closed intervals in *R*. One writes  $A \leq B$  if and only if  $a^L \leq b^L$  and  $a^U \leq b^U$ ,  $A \prec B$  if and only if  $A \leq B$  and  $A \neq B$ , that is, the following (a1), (a2), or (a3) is satisfied:

(a1) 
$$a^{L} < b^{L}$$
 and  $a^{U} \le b^{U}$ ;  
(a2)  $a^{L} \le b^{L}$  and  $a^{U} < b^{U}$ ;  
(a3)  $a^{L} < b^{L}$  and  $a^{U} < b^{U}$ .

*Definition 3* (see [20]). Let  $A = [a^L, a^U]$  and  $B = [b^L, b^U]$  be two closed intervals, the gH-difference of A and B is defined by

$$\begin{bmatrix} a^{L}, a^{U} \end{bmatrix} \ominus_{g} \begin{bmatrix} b^{L}, b^{U} \end{bmatrix}$$
$$= \begin{bmatrix} \min\left(a^{L} - b^{L}, a^{U} - b^{U}\right), \max\left(a^{L} - b^{L}, a^{U} - b^{U}\right) \end{bmatrix}.$$
(3)

For example,  $[1,3] \ominus_g [0,3] = [0,1], [0,3] \ominus_g [1,3] = [-1,0]$ . And  $a - b = [a,a] \ominus_g [b,b] = [a-b,a-b] = a-b$ .

**Proposition 4.** (i) For every  $A, B \in \mathcal{F}$ ,  $A \ominus_g B$  always exists and  $A \ominus_a B \in \mathcal{F}$ .

(ii) 
$$A \ominus_g B \leq 0$$
 if and only if  $A \leq B$ .

#### 3. Interval-Valued Univex Functions

Definition 5 (interval-valued function). The function  $f : \Omega \to \mathcal{F}$  is called an interval-valued function, where  $\Omega \subseteq \mathbb{R}^n$ . Then,  $f(\mathbf{x}) = f(x_1, \dots, x_n)$  is a closed interval in  $\mathbb{R}$  for each  $\mathbf{x} \in \mathbb{R}^n$ , and  $f(\mathbf{x})$  can be also written as  $f(\mathbf{x}) = [f^L(\mathbf{x}), f^U(\mathbf{x})]$ , where  $f^L(\mathbf{x})$  and  $f^U(\mathbf{x})$  are two real-valued functions defined on  $\mathbb{R}^n$  and satisfy  $f^L(\mathbf{x}) \leq f^U(\mathbf{x})$  for every  $\mathbf{x} \in \Omega$ .

Definition 6 (continuity of an interval-valued function). The function  $f: \Omega \subseteq \mathbb{R}^n \to \mathcal{F}$  is said to be continuous at  $x \in \Omega$  if both  $f^L(\mathbf{x})$  and  $f^U(\mathbf{x})$  are continuous functions of  $\mathbf{x}$ .

The concept of gH-derivative of a function  $f : (a, b) \rightarrow \mathcal{I}$  is defined in [19].

*Definition 7.* Let  $x_0 \in (a, b)$  and h be such that  $x_0 + h \in (a, b)$ , then the gH-derivative of a function  $f : (a, b) \to \mathcal{F}$  at  $x_0$  is defined as

$$f'(x_0) = \lim_{x \to 0} \left[ f(x_0 + h) \Theta_g f(x_0) \right].$$
(4)

If  $f'(x_0) \in \mathcal{F}$  exists, then we say that f is generalized Hukuhara differentiable (gH-differentiable, for short) at  $x_0$ . Moreover, [21] also proved the following theorem.

**Theorem 8.** Let  $f : (a,b) \rightarrow \mathcal{F}$  be such that  $f(x) = \int f(x) dx$ 

 $[f^{L}(x), f^{U}(x)]$ . The function f(x) is gH-differentiable if and only if  $f^{L}(x)$  and  $f^{U}(x)$  are differentiable real-valued functions. Furthermore,

$$f'(x) = \left[\min\left\{\left(f^{L}\right)'(x), \left(f^{U}\right)'(x)\right\}, \\ \max\left\{\left(f^{L}\right)'(x), \left(f^{U}\right)'(x)\right\}\right].$$
(5)

Definition 9 (gradient of an interval-valued function). Let  $f(\mathbf{x})$  be an interval-valued function defined on  $\Omega$ , where  $\Omega$  is an open subset of  $\mathbb{R}^n$ . Let  $D_{x_i}$  (i = 1, 2, ..., n) stand for the partial differentiation with respect to the *i*th variable  $x_i$ . Assume that  $f^L(\mathbf{x})$  and  $f^U(\mathbf{x})$  have continuous partial derivatives so that  $D_{x_i}f^L(\mathbf{x})$  and  $D_{x_i}f^U(\mathbf{x})$  are continuous. For i = 1, 2, ..., n, define

$$D_{x_i} f(\mathbf{x}) = \left[ \min \left( D_{x_i} f^L(\mathbf{x}), D_{x_i} f^U(\mathbf{x}) \right), \\ \max \left( D_{x_i} f^L(\mathbf{x}), D_{x_i} f^U(\mathbf{x}) \right) \right].$$
(6)

We will say that  $f(\mathbf{x})$  is differentiable at  $\mathbf{x}$ , and we write

$$\nabla f(\mathbf{x}) = \left( D_{x_1} f(\mathbf{x}), D_{x_2} f(\mathbf{x}), \dots, D_{x_n} f(\mathbf{x}) \right)^t.$$
(7)

We call  $\nabla f(\mathbf{x})$  the gradient of the interval-valued univex function at  $\mathbf{x}$ .

*Example 10.* Let  $f : \mathbf{R}^2 \to \mathcal{F}$  defined by  $f(\mathbf{x}) = [x_1^2 + x_2^2, 2x_1^2 + 2x_2^2 + 3]$ . So  $f^L(\mathbf{x}) = x_1^2 + x_2^2$  and  $f^U(\mathbf{x}) = 2x_1^2 + 2x_2^2 + 3$ .  $D_{x_1}f^L(\mathbf{x}) = 2x_1$ ,  $D_{x_2}f^L(\mathbf{x}) = 2x_2$ ,  $D_{x_1}f^U(\mathbf{x}) = 4x_1$ ,  $D_{x_2}f^U(\mathbf{x}) = 4x_2$ . Thus,

$$D_{x_1} f(\mathbf{x}) = \begin{cases} [2x_1, 4x_1] & \text{if } x_1 \ge 0, \\ [4x_1, 2x_1] & \text{if } x_1 < 0, \end{cases}$$

$$D_{x_2} f(\mathbf{x}) = \begin{cases} [2x_2, 4x_2] & \text{if } x_2 \ge 0, \\ [4x_2, 2x_2] & \text{if } x_2 < 0. \end{cases}$$
(8)

Thus,

$$\nabla f (\mathbf{x}) = \begin{cases} \left( \begin{bmatrix} 2x_1, 4x_1 \end{bmatrix}, \begin{bmatrix} 2x_2, 4x_2 \end{bmatrix} \right)^t & \text{if } x_1 \ge 0, x_2 \ge 0, \\ \left( \begin{bmatrix} 2x_1, 4x_1 \end{bmatrix}, \begin{bmatrix} 4x_2, 2x_2 \end{bmatrix} \right)^t & \text{if } x_1 \ge 0, x_2 < 0, \\ \left( \begin{bmatrix} 4x_1, 2x_1 \end{bmatrix}, \begin{bmatrix} 2x_2, 4x_2 \end{bmatrix} \right)^t & \text{if } x_1 < 0, x_2 \ge 0, \\ \left( \begin{bmatrix} 4x_1, 2x_1 \end{bmatrix}, \begin{bmatrix} 4x_2, 4x_2 \end{bmatrix} \right)^t & \text{if } x_1 < 0, x_2 < 0. \end{cases}$$

$$(9)$$

Further,

$$\nabla^{L} f(\mathbf{x}) = \begin{cases} (2x_{1}, 2x_{2})^{t} & \text{if } x_{1} \ge 0, x_{2} \ge 0, \\ (2x_{1}, 4x_{2})^{t} & \text{if } x_{1} \ge 0, x_{2} < 0, \\ (4x_{1}, 2x_{2})^{t} & \text{if } x_{1} < 0, x_{2} \ge 0, \\ (4x_{1}, 4x_{2})^{t} & \text{if } x_{1} < 0, x_{2} < 0. \end{cases}$$
(10)  
$$\nabla^{U} f(\mathbf{x}) = \begin{cases} (4x_{1}, 4x_{2})^{t} & \text{if } x_{1} \ge 0, x_{2} \ge 0, \\ (4x_{1}, 2x_{2})^{t} & \text{if } x_{1} \ge 0, x_{2} \ge 0, \\ (2x_{1}, 4x_{2})^{t} & \text{if } x_{1} < 0, x_{2} < 0, \\ (2x_{1}, 4x_{2})^{t} & \text{if } x_{1} < 0, x_{2} < 0. \end{cases}$$

*Remark 11.* If  $f^L = f^U$ , then  $\nabla f(\mathbf{x})$  of interval-valued functions is the extension of  $\nabla f(\mathbf{x})$ , where  $f : \Omega \to R$ .

The concept of convexity plays an important role in the optimization theory. In recent years, the concept of convexity has been generalized in several directions by using novel and innovative techniques. An important generalization of convex functions is the introduction of univex functions, which was introduced by Bector et al. [15].

Let *K* be a nonempty open set in  $\mathbb{R}^n$ , and let  $f: K \to \mathbb{R}$ ,  $\eta: K \times K \to \mathbb{R}^n$ ,  $\Phi: \mathbb{R} \to \mathbb{R}$ , and  $b: K \times K \times [0,1] \to \mathbb{R}^+$ ,  $b = b(\mathbf{x}, \mathbf{y}, \lambda)$ . If the function *f* is differentiable, then *b* does not depend on  $\lambda$ ; see [12] or [15].

*Definition 12.* A differentiable real-valued function f is said to be univex at  $\mathbf{y} \in K$  with respect to  $\eta$ ,  $\Phi$ , b if for all  $\mathbf{x} \in K$ 

$$b(\mathbf{x}, \mathbf{y}) \Phi \left[ f(\mathbf{x}) - f(\mathbf{y}) \right] \ge \eta^{t} \left( \mathbf{x}, \mathbf{y} \right) \nabla f(\mathbf{y}).$$
(11)

Let *K* be a nonempty open set in  $\mathbb{R}^n$ , and let  $f : K \to \mathcal{F}$ be an interval-valued function,  $\eta : K \times K \to \mathbb{R}^n$ ,  $\Phi : \mathcal{F} \to \mathcal{F}$ , and  $b : K \times K \times [0, 1] \to \mathbb{R}^+$ ,  $b = b(\mathbf{x}, \mathbf{y}, \lambda)$ .

*Definition 13* (interval-valued univex function). A differentiable interval-valued function f is said to be univex at  $\mathbf{y} \in K$ with respect to  $\eta$ ,  $\Phi$ , b if for all  $\mathbf{x} \in K$ 

$$b(\mathbf{x}, \mathbf{y}) \Phi \left[ f(\mathbf{x}) \ominus_{g} f(\mathbf{y}) \right] \succeq \eta^{t}(\mathbf{x}, \mathbf{y}) \nabla f(\mathbf{y}).$$
(12)

*Remark 14.* (i) An interval-valued univex function is the extension of a univex function by  $f^L = f^U$ .

(ii)  $\Phi : \mathscr{F} \to \mathscr{F}$  could be deduced from  $\phi : R \to R$  by  $\Phi(A) := \{y : \exists x \in A, \phi(x) = y, y \in R\}.$ 

*Example 15.* Consider the real-valued function  $\phi_1$  given by  $\phi_1(x) = x + 1, x \in R$ , then we can obtain  $\Phi_1([a^L, a^U]) = [a^L + 1, a^U + 1]$ . If  $\phi_2(x) = |x|, x \in R$ . Then

$$\Phi_{2}\left(\left[a^{L}, a^{U}\right]\right) = \begin{cases} \left[a^{L}, a^{U}\right] & \text{if } a^{L} \ge 0, \\ \left[-a^{U}, -a^{L}\right] & \text{if } a^{U} \le 0, \\ \left[0, \max\left(-a^{L}, a^{U}\right)\right] & \text{if } a^{L} < 0, a^{U} \ge 0. \end{cases}$$
(13)

*Example 16.* Let  $f(x) = [x^2, 2x^2 + 3]$ ,  $x \in R$ , b = 1,  $\eta(x, y) = x - y$ ,  $\Phi = \Phi_2$ , then f(x) is univex with respect to b,  $\eta$ , and  $\Phi$ .

*Example 17.* Let  $f(x) = [x^3, x^3 + 1], x \in R$ ,

$$b(x, y) = \begin{cases} \frac{y^2}{x - y} & \text{if } x \ge y, \\ 0 & \text{if } x \le y, \end{cases}$$
(14)  
$$\eta(x, y) = \begin{cases} x^2 + y^2 + xy & \text{if } x \ge y, \\ x - y & \text{if } x \le y. \end{cases}$$

Let  $\Phi : \mathscr{F} \to \mathscr{F}$  be defined by  $\Phi([a^L, a^U]) = 3[a^L, a^U]$ , then f(x) is univex with respect to  $b, \eta$  and  $\Phi$ .

#### 4. Optimality Criteria

Let  $f(\mathbf{x}), g_1(\mathbf{x}), \ldots, g_m(\mathbf{x})$  be differentiable interval-valued functions defined on a nonempty open set  $X \subseteq \mathbb{R}^n$ . Throughout this paper we consider the following primal problem (P):

min 
$$f(\mathbf{x})$$
  
s.t.  $q(\mathbf{x}) \leq 0, \quad i = 1, 2, \dots, m.$  (P)

Let  $P := {\mathbf{x} \in X : g(\mathbf{x}) \le 0, i = 1, 2, ..., m}$ . We say  $\mathbf{x}^*$  is an optimal solution of (P) if  $f(\mathbf{x}) \ge f(\mathbf{x}^*)$  for all P-feasible  $\mathbf{x}$ . In this section, we obtain sufficient optimality conditions for a feasible solution  $\mathbf{x}^*$  to be efficient or properly efficient for (P) in the form of the following theorems.

#### **Theorem 18.** Let $\mathbf{x}^*$ be P-feasible. Suppose that

(i) there exist  $\eta$ ,  $\Phi_0$ ,  $b_0$ ,  $\Phi_i$ ,  $b_i$ , i = 1, 2, ..., m such that

$$b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right)\Phi_{0}\left[f\left(\mathbf{x}\right)\ominus_{g}f\left(\mathbf{x}^{*}\right)\right] \geq \eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right), \quad (15)$$

$$-b_{i}(\mathbf{x},\mathbf{x}^{*})\Phi_{i}[g_{i}(\mathbf{x}^{*})] \succeq \eta^{*}(\mathbf{x},\mathbf{x}^{*})\nabla g_{i}(\mathbf{x}^{*})$$
(16)

for all feasible **x**;

(ii) there exist  $\mathbf{y}^* = (y_1, y_2, \dots, y_m)^t \in \mathbb{R}^m$  such that

$$\nabla f\left(\mathbf{x}^{*}\right) + \sum_{i=1}^{m} y_{i} \nabla g\left(\mathbf{x}^{*}\right) = 0, \qquad (17)$$

$$\mathbf{y}^* \ge \mathbf{0}.\tag{18}$$

Further, suppose that

$$\Phi_0(A) \succeq 0 \Longrightarrow A \succeq 0, \tag{19}$$

$$A \le 0 \Longrightarrow \Phi_i(A) \ge 0, \tag{20}$$

$$b_0(\mathbf{x}, \mathbf{x}^*) > 0, \qquad b_i(\mathbf{x}, \mathbf{x}^*) > 0$$
 (21)

for all feasible **x**. Then,  $\mathbf{x}^*$  is an optimal solution of (P).

Proof. Let x be P-feasible. Then,

$$g_i(\mathbf{x}) \le 0. \tag{22}$$

From (20), we conclude that

$$\Phi_i \left[ g_i \left( \mathbf{x} \right) \right] \succeq 0. \tag{23}$$

Thus,

$$\Phi_{i}^{L}\left[g_{i}\left(\mathbf{x}\right)\right] \geq 0,$$

$$\Phi_{i}^{U}\left[g_{i}\left(\mathbf{x}\right)\right] \geq 0.$$
(24)

By (15) and Definition 2, we have

$$\left\{ b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right)\Phi_{0}\left[f\left(\mathbf{x}\right)\ominus_{g}f\left(\mathbf{x}^{*}\right)\right]\right\}^{L} \geq \left\{ \eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{L}, \\ \left\{ b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right)\Phi_{0}\left[f\left(\mathbf{x}\right)\ominus_{g}f\left(\mathbf{x}^{*}\right)\right]\right\}^{U} \geq \left\{ \eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{U}.$$

$$(25)$$

From (17),

$$\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)+\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)=0.$$
 (26)

It follows from Definition 2 that

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{L}+\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{L}=0,$$
$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{U}+\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{U}=0.$$
(27)

It is equivalent to

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{L} = -\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{L},$$

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{U} = -\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{U}.$$
(28)

From (16), we have

$$\{-b_{i}\left(\mathbf{x},\mathbf{x}^{*}\right)\Phi_{i}\left[g_{i}\left(\mathbf{x}^{*}\right)\right]\}^{L} \geq \left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla g_{i}\left(\mathbf{x}^{*}\right)\right\}^{L},$$

$$\{-b_{i}\left(\mathbf{x},\mathbf{x}^{*}\right)\Phi_{i}\left[g_{i}\left(\mathbf{x}^{*}\right)\right]\}^{U} \geq \left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla g_{i}\left(\mathbf{x}^{*}\right)\right\}^{U}.$$

$$(29)$$

From Definition 1, we have

$$-\{b_{i}(\mathbf{x},\mathbf{x}^{*})\Phi_{i}[g_{i}(\mathbf{x}^{*})]\}^{U} \geq \{\eta^{t}(\mathbf{x},\mathbf{x}^{*})\nabla g_{i}(\mathbf{x}^{*})\}^{L},$$

$$-\{b_{i}(\mathbf{x},\mathbf{x}^{*})\Phi_{i}[g_{i}(\mathbf{x}^{*})]\}^{L} \geq \{\eta^{t}(\mathbf{x},\mathbf{x}^{*})\nabla g_{i}(\mathbf{x}^{*})\}^{U}.$$

$$(30)$$

Thus,

$$\left\{ b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right) \Phi_{0}\left[f\left(\mathbf{x}\right) \ominus_{g}f\left(\mathbf{x}^{*}\right)\right] \right\}^{L}$$

$$\geq \left\{ \eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right) \nabla f\left(\mathbf{x}^{*}\right) \right\}^{L}$$

$$= -\left\{ \eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right) \sum_{i=1}^{m} y_{i} \nabla g\left(\mathbf{x}^{*}\right) \right\}^{L}$$

$$\geq \sum_{i=1}^{m} \left\{ b_{i}\left(\mathbf{x},\mathbf{x}^{*}\right) \Phi_{i}\left[g_{i}\left(\mathbf{x}^{*}\right)\right] \right\}^{U}$$

$$\geq \sum_{i=1}^{m} \left\{ b_{i}\left(\mathbf{x},\mathbf{x}^{*}\right) \Phi_{i}\left[g_{i}\left(\mathbf{x}^{*}\right)\right] \right\}^{L}$$

$$\geq 0,$$

$$\left\{ b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right) \Phi_{0}\left[f\left(\mathbf{x}\right) \ominus_{g}f\left(\mathbf{x}^{*}\right)\right] \right\}^{L}$$

$$\geq \left\{ b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right) \Phi_{0}\left[f\left(\mathbf{x}\right) \ominus_{g}f\left(\mathbf{x}^{*}\right)\right] \right\}^{L}$$

$$\geq 0.$$

$$\left\{ b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right) \Phi_{0}\left[f\left(\mathbf{x}\right) \ominus_{g}f\left(\mathbf{x}^{*}\right)\right] \right\}^{L}$$

$$\geq 0.$$

So,

$$b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right)\Phi_{0}\left[f\left(\mathbf{x}\right)\ominus_{g}f\left(\mathbf{x}^{*}\right)\right]\geq0.$$
(32)

From (21), it follows that

$$\Phi_0\left[f\left(\mathbf{x}\right)\ominus_g f\left(\mathbf{x}^*\right)\right] \succeq 0.$$
(33)

By (19),

$$f(\mathbf{x}) \ominus_g f(\mathbf{x}^*) \succeq 0.$$
(34)

From Proposition 4, it follows that

$$f(\mathbf{x}) \succeq f(\mathbf{x}^*). \tag{35}$$

Therefore,  $\mathbf{x}^*$  is an optimal solution of (P).

#### **Theorem 19.** Let $\mathbf{x}^*$ be P-feasible. Suppose that

(i) there exist 
$$\eta$$
,  $\Phi_0$ ,  $b_0$ ,  $\Phi_i$ ,  $b_i$ ,  $i = 1, 2, ..., m$  such that  
 $\eta^t(\mathbf{x}, \mathbf{x}^*) \nabla f(\mathbf{x}^*) \ge 0 \Longrightarrow b_0(\mathbf{x}, \mathbf{x}^*) \Phi_0[f(\mathbf{x}) \ominus_g f(\mathbf{x}^*)] \ge 0$ ,

$$-b_{i}(\mathbf{x},\mathbf{x}^{*})\Phi_{i}[g_{i}(\mathbf{x}^{*})] \leq 0 \Longrightarrow \eta^{t}(\mathbf{x},\mathbf{x}^{*})\nabla g_{i}(\mathbf{x}^{*}) \leq 0$$
(37)

for all feasible **x**;

(ii) there exist  $\mathbf{y}^* = (y_1, y_2, \dots, y_m)^t \in \mathbb{R}^m$  such that

$$\nabla f\left(\mathbf{x}^{*}\right) + \sum_{i=1}^{m} y_{i} \nabla g\left(\mathbf{x}^{*}\right) = 0, \qquad (38)$$

$$\mathbf{y}^* \ge \mathbf{0}. \tag{39}$$

Further, suppose that

$$\Phi_0(A) \succeq 0 \Longrightarrow A \succeq 0, \tag{40}$$

$$A \le 0 \Longrightarrow \Phi_i(A) \ge 0, \tag{41}$$

 $b_0(\mathbf{x}, \mathbf{x}^*) > 0, \qquad b_i(\mathbf{x}, \mathbf{x}^*) > 0$  (42)

for all feasible  $\mathbf{x}$ . Then,  $\mathbf{x}^*$  is an optimal solution of (P).

*Proof.* Let **x** be P-feasible. Then,  $g_i(\mathbf{x}^*) \leq 0$ , from (41), we obtain that

$$\Phi_i\left[g_i\left(\mathbf{x}^*\right)\right] \succeq 0. \tag{43}$$

So,

$$-b_i\left(\mathbf{x},\mathbf{x}^*\right)\Phi_i\left[g_i\left(\mathbf{x}^*\right)\right] \le 0.$$
(44)

By (37),

$$\eta^{t}(\mathbf{x}, \mathbf{x}^{*}) \nabla g_{i}(\mathbf{x}^{*}) \leq 0.$$
(45)

Thus,

$$\sum_{i=1}^{m} y_{i} \eta^{t} \left( \mathbf{x}, \mathbf{x}^{*} \right) \nabla g_{i} \left( \mathbf{x}^{*} \right) \succeq 0.$$
(46)

Then, we have

$$-\left\{\sum_{i=1}^{m} y_{i} \eta^{t}\left(\mathbf{x}, \mathbf{x}^{*}\right) \nabla g_{i}\left(\mathbf{x}^{*}\right)\right\}^{U}$$

$$=\left\{-\sum_{i=1}^{m} y_{i} \eta^{t}\left(\mathbf{x}, \mathbf{x}^{*}\right) \nabla g_{i}\left(\mathbf{x}^{*}\right)\right\}^{L} \ge 0,$$

$$-\left\{\sum_{i=1}^{m} y_{i} \eta^{t}\left(\mathbf{x}, \mathbf{x}^{*}\right) \nabla g_{i}\left(\mathbf{x}^{*}\right)\right\}^{L}$$

$$=\left\{-\sum_{i=1}^{m} y_{i} \eta^{t}\left(\mathbf{x}, \mathbf{x}^{*}\right) \nabla g_{i}\left(\mathbf{x}^{*}\right)\right\}^{U} \ge 0.$$
(47)

From (38) and Definition 2, it follows that

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{L}+\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{L}=0,$$

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{U}+\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{U}=0.$$
(48)

It is equivalent to

(36)

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{L} = -\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{L},$$

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right)\right\}^{U} = -\left\{\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\sum_{i=1}^{m}y_{i}\nabla g\left(\mathbf{x}^{*}\right)\right\}^{U}.$$
(49)

Therefore,

$$\left\{ \eta^{t} \left( \mathbf{x}, \mathbf{x}^{*} \right) \nabla f \left( \mathbf{x}^{*} \right) \right\}^{L} \geq 0,$$

$$\left\{ \eta^{t} \left( \mathbf{x}, \mathbf{x}^{*} \right) \nabla f \left( \mathbf{x}^{*} \right) \right\}^{U} \geq 0.$$

$$(50)$$

From Definition 2, we obtain that

$$\eta^{t}\left(\mathbf{x},\mathbf{x}^{*}\right)\nabla f\left(\mathbf{x}^{*}\right) \succeq 0.$$
(51)

By (36),

$$b_{0}\left(\mathbf{x},\mathbf{x}^{*}\right)\Phi_{0}\left[f\left(\mathbf{x}\right)\ominus_{g}f\left(\mathbf{x}^{*}\right)\right]\geq0.$$
(52)

Then, from (40) and (42), we have

$$f(\mathbf{x}) \ominus_g f(\mathbf{x}^*) \succeq 0.$$
(53)

From Proposition 4, it follows that

max  $f(\mathbf{u})$ 

$$f(\mathbf{x}) \succeq f(\mathbf{x}^*). \tag{54}$$

Therefore,  $\mathbf{x}^*$  is an optimal solution of (P).

#### 5. Duality

Consider the following:

s.t. 
$$\nabla f(\mathbf{u}) + \sum_{i=1}^{m} y_i \nabla g_i(\mathbf{u}) = 0,$$
 (D)  
 $y_i \nabla g_i(\mathbf{u}) \ge 0$   
 $y_i \ge 0.$ 

**Theorem 20** (weak duality). Let **x** be P-feasible, and let  $(\mathbf{u}, \mathbf{y})$  be D-feasible. Assume that there exist  $\eta$ ,  $\Phi_0$ ,  $b_0$ ,  $\Phi_i$ ,  $b_i$ , i = 1, 2, ..., m such that

$$b_{0}(\mathbf{x}, \mathbf{u}) \Phi_{0}\left[f(\mathbf{x}) \ominus_{g} f(\mathbf{u})\right] \geq \eta^{t}(\mathbf{x}, \mathbf{u}) \nabla f(\mathbf{u}),$$
  
$$-b_{i}(\mathbf{x}, \mathbf{u}) \Phi_{i}\left[g_{i}(\mathbf{u})\right] \geq \eta^{t}(\mathbf{x}, \mathbf{u}) \nabla g_{i}(\mathbf{u})$$
(55)

at **u**;

$$\Phi_0(A) \ge 0 \Longrightarrow A \ge 0,$$
  

$$b_0(\mathbf{x}, \mathbf{u}) > 0, \qquad b_i(\mathbf{x}, \mathbf{u}) \ge 0$$
(56)

and  $\sum_{i=1}^{m} b_i(\mathbf{x}, \mathbf{u}) y_i \Phi_i(g_i(\mathbf{u})) \ge 0$ . Then,  $f(\mathbf{x}) \ge f(\mathbf{u})$ .

*Proof.* It is similar to the proof of Theorem 18.  $\Box$ 

**Theorem 21** (weak duality). Let **x** be P-feasible, and let  $(\mathbf{u}, \mathbf{y})$  be D-feasible. Assume that there exist  $\eta$ ,  $\Phi_0$ ,  $b_0$ ,  $\Phi_1$ ,  $b_1$  such that

$$\eta^{t}(\mathbf{x}, \mathbf{u}) \nabla f(\mathbf{u}) \succeq 0 \Longrightarrow b_{0}(\mathbf{x}, \mathbf{u}) \Phi_{0}\left[f(\mathbf{x}) \ominus_{g} f(\mathbf{u})\right] \succeq 0,$$
(57)

$$-b_{1}(\mathbf{x},\mathbf{u})\Phi_{1}\left[\sum_{i=1}^{m}y_{i}g_{i}(\mathbf{u})\right] \leq 0 \Longrightarrow \sum_{i=1}^{m}y_{i}\eta^{t}(\mathbf{x},\mathbf{u})\nabla g_{i}(\mathbf{u}) \leq 0$$
(58)

at **u**;

$$\Phi_0(A) \ge 0 \Longrightarrow A \ge 0, \tag{59}$$

$$A \succeq 0 \Longrightarrow \Phi_1(A) \succeq 0, \tag{60}$$

$$b_0(\mathbf{x}, \mathbf{u}) > 0, \qquad b_1(\mathbf{x}, \mathbf{u}) \ge 0.$$
 (61)

Then,  $f(\mathbf{x}) \succeq f(\mathbf{u})$ .

*Proof.* Since  $(\mathbf{u}, \mathbf{y})$  is D-feasible, then  $y_i^t \nabla g_i(\mathbf{u}) \ge 0$ , from (60) and (61),

$$-b_{1}(\mathbf{x},\mathbf{u})\Phi_{1}\left[\sum_{i=1}^{m}y_{i}g_{i}(\mathbf{u})\right] \leq 0.$$
(62)

Then, we have

$$\sum_{i=1}^{m} y_i \eta^t \left( \mathbf{x}, \mathbf{u} \right) \nabla g_i \left( \mathbf{u} \right) \le 0.$$
(63)

Thus,

$$-\sum_{i=1}^{m} y_{i} \eta^{t} (\mathbf{x}, \mathbf{u}) \nabla g_{i} (\mathbf{u}) \geq 0,$$

$$-\left\{\sum_{i=1}^{m} y_{i} \eta^{t} (\mathbf{x}, \mathbf{u}) \nabla g_{i} (\mathbf{u})\right\}^{U}$$

$$=\left\{-\sum_{i=1}^{m} y_{i} \eta^{t} (\mathbf{x}, \mathbf{u}) \nabla g_{i} (\mathbf{u})\right\}^{L} \geq 0, \quad (64)$$

$$-\left\{\sum_{i=1}^{m} y_{i} \eta^{t} (\mathbf{x}, \mathbf{u}) \nabla g_{i} (\mathbf{u})\right\}^{L}$$

$$=\left\{-\sum_{i=1}^{m} y_{i} \eta^{t} (\mathbf{x}, \mathbf{u}) \nabla g_{i} (\mathbf{u})\right\}^{U} \geq 0.$$

Since  $(\mathbf{u}, \mathbf{y})$  is D-feasible we can obtain that,

$$\nabla f(\mathbf{u}) + \sum_{i=1}^{m} y_i \nabla g_i(\mathbf{u}) = 0.$$
(65)

So,

$$\eta^{t}(\mathbf{x},\mathbf{u})\nabla f(\mathbf{u}) + \eta^{t}(\mathbf{x},\mathbf{u})\sum_{i=1}^{m} y_{i}\nabla g_{i}(\mathbf{u}) = 0.$$
(66)

By Definition 2, it follows that

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\nabla f\left(\mathbf{u}\right)\right\}^{L}+\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\sum_{i=1}^{m}y_{i}\nabla g_{i}\left(\mathbf{u}\right)\right\}^{L}=0,$$

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\nabla f\left(\mathbf{u}\right)\right\}^{U}+\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\sum_{i=1}^{m}y_{i}\nabla g_{i}\left(\mathbf{u}\right)\right\}^{U}=0.$$
(67)

Therefore,

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\nabla f\left(\mathbf{u}\right)\right\}^{L} = -\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\sum_{i=1}^{m}y_{i}\nabla g_{i}\left(\mathbf{u}\right)\right\}^{L} \ge 0,$$

$$\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\nabla f\left(\mathbf{u}\right)\right\}^{U} = -\left\{\eta^{t}\left(\mathbf{x},\mathbf{u}\right)\sum_{i=1}^{m}y_{i}\nabla g_{i}\left(\mathbf{u}\right)\right\}^{U} \ge 0.$$
(68)

Then,

$$\eta^t \left( \mathbf{x}, \mathbf{u} \right) \nabla f \left( \mathbf{u} \right) \succeq 0.$$
(69)

By (57),

$$b_{0}(\mathbf{x}, \mathbf{u}) \Phi_{0}\left[f(\mathbf{x}) \ominus_{g} f(\mathbf{u})\right] \succeq 0.$$
(70)

From (59) and (61),

$$f(\mathbf{x}) \ominus_g f(\mathbf{u}) \succeq 0. \tag{71}$$

thus,

$$f(\mathbf{x}) \succeq f(\mathbf{u}). \tag{72}$$

**Theorem 22** (strong duality). If  $\mathbf{x}^*$  is P-optimal and a constraint qualification is satisfied at  $\mathbf{x}^*$ , then there exists  $\mathbf{y}^* = (y_1, y_2, \ldots, y_m)^t \in \mathbb{R}^m$  such that  $(\mathbf{x}^*, \mathbf{y}^*)$  is D-feasible and the values of the objective functions for (P) and (D) are equal at  $\mathbf{x}^*$  and  $(\mathbf{x}^*, \mathbf{y}^*)$ , respectively. Furthermore, if for all P-feasible  $\mathbf{x}$  and D-feasible  $(\mathbf{u}, \mathbf{y})$ , the hypotheses of Theorem 19 are satisfied, then  $(\mathbf{x}^*, \mathbf{y}^*)$  is D-optimal.

*Proof.* Since a constraint qualification is satisfied at  $\mathbf{x}^*$ , there exists  $\mathbf{y}^* \in \mathbb{R}^m$  such that the following Kuhn-Tucker conditions are satisfied:

$$\nabla f\left(\mathbf{x}^{*}\right) + \sum_{i=1}^{m} y_{i} \nabla g_{i}\left(\mathbf{x}^{*}\right) = 0,$$

$$\sum_{i=1}^{m} y_{i} \nabla g_{i}\left(\mathbf{x}^{*}\right) = 0,$$

$$y_{i} \ge 0.$$
(73)

Therefore,  $(\mathbf{x}^*, \mathbf{y}^*)$  is D-feasible.

Suppose that  $(\mathbf{x}^*, \mathbf{y}^*)$  is not D-optimal. Then, there exists a D-feasible  $(\mathbf{u}, \mathbf{y})$  such that  $f(\mathbf{u}) \succ f(\mathbf{x}^*)$ . This contradicts Theorem 20. Therefore,  $(\mathbf{x}^*, \mathbf{y}^*)$  is D-optimal.

#### 6. Numerical Example

Consider the following example:

minimize 
$$f(\mathbf{x}) = [x_1 - \sin(x_2) + 1, x_1 - \sin(x_2) + 3]$$
  
subject to  $g_1(\mathbf{x})$   
 $= [\sin(x_1) - 4\sin(x_2) - 2, 
 $\sin(x_1) - 4\sin(x_2)] \le 0,$   
 $g_2(\mathbf{x}) = [2\sin(x_1) + 7\sin(x_2) + x_1 - 7, 
 $2\sin(x_1) + 7\sin(x_2) + x_1 - 6]$   
 $\le 0,$   
 $g_3(\mathbf{x}) = [2x_1 + 2x_2 - 5, 2x_1 + 2x_2 - 3] \le 0,$   
 $g_4(\mathbf{x}) = [4x_1^2 + 4x_2^2 - 12, 4x_1^2 + 4x_2^2 - 9] \le 0,$   
 $g_5(\mathbf{x}) = [-\sin(x_1) - 1, -\sin(x_1)] \le 0,$   
 $g_6(\mathbf{x}) = [-\sin(x_2) - 1, -\sin(x_2)] \le 0.$ 
(74)$$ 

Note that the interval-valued objective function is univex with respect to b = 1,  $\eta(\mathbf{x}, \mathbf{u}) = \mathbf{x} - \mathbf{u}$ ,  $\Phi([a^L, a^U]) = [a^L, a^U]$ , and every  $g_i$  (i = 1, 2, ..., 6) is univex with respect to b = 1,  $\Phi([a^L, a^U]) = [a^L, a^U]$ 

$$\eta\left(\mathbf{x},\mathbf{u}\right) = \left(\frac{\sin x_1 - \sin u_1}{\cos u_1}, \frac{\sin x_2 - \sin u_2}{\cos u_2}\right)^T, \quad (75)$$

where **x** =  $(x_1, x_2)^T$  and **u** =  $(u_1, u_2)^T$ .

It is easy to see that the problem satisfies the assumptions of Theorem 18. Then,

$$(1, -\cos x_2)^T + \mu_1(\cos x_1, -4\cos x_2)^T + \mu_2(2\cos x_1 + 1, 7\cos x_2)^T + \mu_3(2, 2)^T + \mu_4(8x_1, 8x_2)^T + \mu_5(-\cos x_1, 0)^T + \mu_6(0, -\cos x_2)^T = (0, 0)^T.$$
(76)

After some algebraic calculations, we obtain that  $\mathbf{x}^* = (0, \sin^{-1}(6/7))^T$  and  $\mathbf{u}^* = (0, 1/7, 0, 0, 10/7, 0)^T$ . Therefore,  $\mathbf{x}^*$  is a solution.

#### Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 60974082), and the Science Plan Foundation of the Education Bureau of Shaanxi Province (nos. 11JK1051, 2013JK1098, 2013JK1130, and 2013JK1182).

#### References

- C. Jiang, X. Han, G. R. Liu, and G. P. Liu, "A nonlinear interval number programming method for uncertain optimization problems," *European Journal of Operational Research*, vol. 188, no. 1, pp. 1–13, 2008.
- [2] S. Chanas and D. Kuchta, "Multiobjective programming in optimization of interval objective functions—a generalized approach," *European Journal of Operational Research*, vol. 94, no. 3, pp. 594–598, 1996.
- [3] S.-T. Liu, "Posynomial geometric programming with interval exponents and coefficients," *European Journal of Operational Research*, vol. 186, no. 1, pp. 7–27, 2008.
- [4] H.-C. Wu, "The Karush-Kuhn-Tucker optimality conditions in an optimization problem with interval-valued objective function," *European Journal of Operational Research*, vol. 176, no. 1, pp. 46–59, 2007.
- [5] H.-C. Wu, "The Karush-Kuhn-Tucker optimality conditions in multiobjective programming problems with interval-valued objective functions," *European Journal of Operational Research*, vol. 196, no. 1, pp. 49–60, 2009.
- [6] H.-C. Wu, "On interval-valued nonlinear programming problems," *Journal of Mathematical Analysis and Applications*, vol. 338, no. 1, pp. 299–316, 2008.
- [7] H. C. Wu, "Wolfe duality for interval-valued optimization," *Journal of Optimization Theory and Applications*, vol. 138, no. 3, pp. 497–509, 2008.
- [8] H. C. Wu, "Duality theory for optimization problems with interval-valued objective functions," *Journal of Optimization Theory and Applications*, vol. 144, no. 3, pp. 615–628, 2010.
- [9] H.-C. Wu, "Duality theory in interval-valued linear programming problems," *Journal of Optimization Theory and Applications*, vol. 150, no. 2, pp. 298–316, 2011.
- [10] M. A. Hanson, "On sufficiency of the Kuhn-Tucker conditions," *Journal of Mathematical Analysis and Applications*, vol. 80, no. 2, pp. 545–550, 1981.
- [11] R. N. Kaul, S. K. Suneja, and M. K. Srivastava, "Optimality criteria and duality in multiple-objective optimization involving generalized invexity," *Journal of Optimization Theory and Applications*, vol. 80, no. 3, pp. 465–482, 1994.
- [12] C. R. Bector and C. Singh, "B-vex functions," *Journal of Optimization Theory and Applications*, vol. 71, no. 2, pp. 237–253, 1991.
- [13] C. R. Bector, M. K. Bector, A. Gill, and C. Singh, "Duality for vector valued B-invex programming," in *Generalized Convexity Proceedings*, S. Komlosi, T. Rapcsak, and S. Schaible, Eds., pp. 358–373, Springer, Berlin, Germany, 1992.
- [14] C. R. Bector, M. K. Bector, A. Gill, and C. Singh, "Univex sets, functions and univex nonlinear programming," in *Generalized Convexity Proceedings*, S. Komlosi, T. Rapcsak, and S. Schaible, Eds., pp. 3–18, Springer, Berlin, Germany, 1992.
- [15] C. R. Bector, S. K. Suneja, and S. Gupta, "Univex functions and Univex nonlinear programming," in *Proceedings of the Administrative Sciences Association of Canada*, pp. 115–124, 1992.
- [16] N. G. Rueda, M. A. Hanson, and C. Singh, "Optimality and duality with generalized convexity," *Journal of Optimization Theory and Applications*, vol. 86, no. 2, pp. 491–500, 1995.
- [17] B. Aghezzaf and M. Hachimi, "Generalized invexity and duality in multiobjective programming problems," *Journal of Global Optimization*, vol. 18, no. 1, pp. 91–101, 2000.

- [18] T. R. Gulati, I. Ahmad, and D. Agarwal, "Sufficiency and duality in multiobjective programming under generalized type I functions," *Journal of Optimization Theory and Applications*, vol. 135, no. 3, pp. 411–427, 2007.
- [19] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, UK, 1990.
- [20] L. Stefanini, "A generalization of Hukuhara difference for interval and fuzzy arithmetic," in *Soft Methods for Handling Variability and Imprecision*, D. Dubois, M. A. Lubiano et al., Eds., vol. 48 of *Series on Advances in Soft Computing*, Springer, Berlin, Germany, 2008.
- [21] L. Stefanini and B. Bede, "Generalized Hukuhara differentiability of interval-valued functions and interval differential equations," *Nonlinear Analysis: Theory, Methods and Applications A*, vol. 71, no. 3-4, pp. 1311–1328, 2009.

# **Research** Article

# **Smoothing Techniques and Augmented Lagrangian Method for Recourse Problem of Two-Stage Stochastic Linear Programming**

#### Saeed Ketabchi and Malihe Behboodi-Kahoo

Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Guilan, P.O. Box 416351914 Rasht, Iran

Correspondence should be addressed to Malihe Behboodi-Kahoo; behboodi.m@gmail.com

Received 1 February 2013; Accepted 22 April 2013

Academic Editor: Neal N. Xiong

Copyright © 2013 S. Ketabchi and M. Behboodi-Kahoo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The augmented Lagrangian method can be used for solving recourse problems and obtaining their normal solution in solving two-stage stochastic linear programming problems. The augmented Lagrangian objective function of a stochastic linear problem is not twice differentiable which precludes the use of a Newton method. In this paper, we apply the smoothing techniques and a fast Newton-Armijo algorithm for solving an unconstrained smooth reformulation of this problem. Computational results and comparisons are given to show the effectiveness and speed of the algorithm.

#### 1. Introduction

In stochastic programming, some data are random variables with specific possibility distribution [1], which was first introduced by the designer of linear programming problems, Dantzig, in [2].

In this paper, we consider the following two-stage stochastic linear program (slp) with recourse which involves the calculation of an expectation over a discrete set of scenarios:

$$\min_{x \in X} f(x) = c^T x + \phi(x),$$

$$X = \{x \in \mathbb{R}^n : Ax = b, x \ge 0\},$$
(1)

where

$$\phi(x) = E(Q(x,\omega)) = \sum_{i=1}^{N} Q(x,\omega^{i}) \rho(\omega^{i})$$
(2)

and *E* shows the expectation of function  $Q(x, \omega)$  which depend on the random variable  $\omega$ . The function *Q* is defined as follows:

$$Q(x,\omega) = \min_{y \in \mathbb{R}^{n_2}} \left\{ q(\omega)^T y \mid W(\omega)^T y \ge h(\omega) - T(\omega) x \right\},$$
(3)

where  $A \in \mathbb{R}^{m \times n}$ ,  $c \in \mathbb{R}^n$ , and  $b \in \mathbb{R}^m$ . Also, in the problem (3) vector of coefficients  $q(\cdot) \in \mathbb{R}^{n_2}$ , matrix of coefficients

 $W^T(\cdot) \in \mathbb{R}^{m_2 \times n_2}$ , demand vector  $h(\cdot) \in \mathbb{R}^{m_2}$ , and matrix  $T(\cdot) \in \mathbb{R}^{m_2 \times n}$  depend on the random vector  $\omega$  with support space  $\Omega$ . The problems (1) and (3) are called master and recourse problems of stochastic programming, respectively.

We assume that the problem (3) has a solution for each  $x \in X$  and  $\omega \in \Omega$ .

In general, the recourse function  $\phi(x)$  is not differentiable everywhere. Therefore, the traditional methods use nonsmooth optimization techniques [3-5]. However, in the last decade, it is proposed smoothing method for recourse function in standard form of recourse problem [6-11]. In this paper, we apply a smooth approximation technique to smooth recourse function that the recourse problem has inequality linear constrained. For more explanation see Section 2. The approximated problem is based on the least two-norm solution of recourse problem. This paper considers the augmented Lagrangian method to obtain least two-norm solution (Section 3). For convenience, Euclidean least twonorm solution of linear programming problem is named normal solution. This effective method contains solving an unconstrained quadratic problem which its objective function is not twice differentiable. To apply a fast Newton method we use the soothing technique and replace plus function by an accurate smooth approximation [12, 13]. In Section 4, the smoothing algorithm and the numerical results are presented. Also, concluding remarks are given in Section 5.

We now describe our notation. Let  $a = [a_i]$  be a vector in  $\mathbb{R}^n$ . By  $a_+$  we mean a vector in  $\mathbb{R}^n$  whose *i*th entry is 0 if  $a_i < 0$  and equals  $a_i$  if  $a_i \ge 0$ . By  $A^T$  we mean the transpose of matrix A, and  $\nabla f(x_0)$  is the gradient of f at  $x_0$ . For  $x \in \mathbb{R}^n$ , || x || and  $|| x ||_{\infty}$  denote 2-norm and infinity norm, respectively.

#### 2. Approximation of Recourse Function

As mentioned the objective function of (1) is nondifferentiable. This disadvantage property occurs on the recourse function. In this section, there is an attempt to approximate it to a differentiable function.

Using dual of the problem (3), function  $Q(x, \omega)$  can be written as follows:

$$Q(x,\omega) = \max_{z \in \mathbb{R}^{m_2}} (h(\omega) - T(\omega)x)^T z$$
s.t.  $W(\omega) z = q(\omega), \quad z \ge 0.$ 
(4)

Unlike the linear recourse function, the quadratic recours function is differentiable. Thus in this paper, the approximation is based on the following quadratic problem with helpful properties:

$$Q_{\epsilon}(x,\omega) = \max_{z \in \mathbb{R}^{m_2}} (h(\omega) - T(\omega)x)^T z - \frac{\epsilon}{2} \|z\|^2$$
(5)
s.t.  $W(\omega) z = q(\omega), \quad z \ge 0.$ 

The next theorem shows that, for the sufficiently small  $\epsilon > 0$ , the solution of this problem is the normal solution of the problem (4).

**Theorem 1.** For functions  $Q(x, \omega)$  and  $Q_{\epsilon}(x, \omega)$  introduced in (4) and (5), the following can be presented:

- (a) ∃ε > 0 such that, for each ε ∈ (0, ε], the solution for the problem (5) is the normal solution for the problem (4).
- (b) For each ε > 0, function Q<sub>ε</sub>(x, ω) is differentiable with respect to x.
- (c) The gradient of function  $Q_{\epsilon}(x, \omega)$  at point x is

$$\nabla Q_{\epsilon}(x,\omega) = -T^{T}(\omega) z_{\epsilon}^{*}(x,\omega)$$
(6)

in which  $z_{\epsilon}^{*}(x, \omega)$  is the solution of the problem (5).

*Proof.* To prove (a), refer to [14, 15].

Also, (b) and (c) can be easily proved considering that function  $Q_{\epsilon}(x, \omega)$  is the conjugate of function

$$p(z) = \begin{cases} \frac{\epsilon}{2} \|z\|^2, & z \in Z, \\ \infty, & z \notin Z, \end{cases}$$
(7)

where

$$Z = \left\{ z \in \mathbb{R}^{m_2} : W(\omega) \, z = q(\omega) \, , \, z \ge 0 \right\}, \tag{8}$$

Using the approximated recourse function  $Q_{\epsilon}(x, \omega)$ , we can define a differentiable approximation function to the objective function of (1):

$$f_{\epsilon}(x) = c^{T}x + \sum_{i=1}^{N} Q_{\epsilon}(x, \omega^{i}) \rho(\omega^{i}).$$
(9)

By (6), the gradient of above function exists and is obtained by

$$\nabla f_{\epsilon}(x) = c + \sum_{i=1}^{N} \nabla Q_{\epsilon}(x, \omega^{i}) \rho(\omega^{i})$$

$$= c - \sum_{i=1}^{N} T^{T}(\omega^{i}) z_{\epsilon}^{*}(x, \omega^{i}) \rho(\omega^{i}).$$
(10)

This approximation has paved the way to use the optimization algorithm for master problem (1) in which the objective function is substituted by  $f_{\epsilon}(x)$ 

$$\min_{x \in X} f_{\epsilon}(x) \,. \tag{11}$$

In [7], it is considered slp problem with inequality constrained in master problem and equality constrained in recourse problem. Also, in Theorem 2.3 of [7], it is shown that a solution of the approximated problem is a good approximation to a solution of master problem. Here we can express a similar theorem for the problem (1) by using the similar technique in the proof of Theorem 2.3 in [7].

**Theorem 2.** Consider the problem (1). Then, for any  $x \in X$ , there exists an  $\overline{\epsilon}(x) > 0$  such that for any  $\epsilon \in (0, \overline{\epsilon}(x)]$ 

$$\left|f\left(x\right) - f_{\epsilon}\left(x\right)\right| \le \frac{\epsilon}{2}M,$$
 (12)

where *M* is defined as follows:

$$M = \max_{i=1,2,...,N} \left\| z_{\epsilon}^{*}(x,\omega^{i}) \right\|^{2}.$$
 (13)

Let  $x^*$  be a solution of (1) and  $x^*_{\epsilon}$  a solution of (11). Then, there exists an  $\overline{\epsilon} > 0$  such that for any  $0 < \epsilon \leq \overline{\epsilon}$ 

$$\max\left\{f\left(x_{\epsilon}^{*}\right) - f\left(x^{*}\right), f_{\epsilon}\left(x^{*}\right) - f_{\epsilon}\left(x_{\epsilon}^{*}\right)\right\} \le \frac{\epsilon}{2}M.$$
(14)

*Further, one assumes that* f or  $f_{\epsilon}$  *are strongly convex on* X *with modulus*  $\mu > 0$ *. Then,* 

$$\|\boldsymbol{x}^* - \boldsymbol{x}_{\varepsilon}^*\| \le M \frac{\epsilon}{\mu}.$$
(15)

According to Theorem 1, it can be found that for obtaining the gradient of function  $f_{\epsilon}(x)$  in each iteration, we need the normal solution of N linear programming problems (4). In this paper, the augmented Lagrangian method [17] is used for this purpose.

#### 3. Smooth Approximation and Augmented Lagrangian Method

In the augmented Lagrangian method, the unconstrained maximization problem is solved which gives the project of a point on the solution set of the problem (4).

Assume that  $\hat{z}$  is an arbitrary vector. Consider the problem of finding the least 2-norm projection  $\hat{z}_*$  of  $\hat{z}$  on the solution set  $Z_*$  of the problem (4)

$$\frac{1}{2} \| \hat{z}_* - \hat{z} \|^2 = \min_{z \in Z_*} \frac{1}{2} \| z - \hat{z} \|^2,$$

$$Z_* = \left\{ z \in \mathbb{R}^{m_2} : W(\omega) \, z = q(\omega) \,, \ \xi^T z = Q(x, \omega) \,, \ z \ge 0 \right\}.$$
(16)

In this problem, vector x and random variable  $\omega$  are constants; therefore, for simplicity, this is assumed to be  $\xi = h(\omega) - T(\omega)x$ , and function  $\widehat{Q}(\xi)$  is defined in a way that  $\widehat{Q}(\xi) = Q(x, \omega)$ .

Considering that the objective function of the problem (16) is strictly convex, its solution is unique. Let us introduce the Lagrangian function for the problem (16) as follow:

$$L(z, p, \beta, \widehat{z}, \xi, \omega) = \frac{1}{2} \|z - \widehat{z}\|^{2} + p^{T} (W(\omega) z - q(\omega)) + \beta (\xi^{T} z - \widehat{Q}(\xi)),$$
(17)

where  $p \in \mathbb{R}^{n_2}$  and  $\beta \in \mathbb{R}$  are Lagrangian multipliers and  $\xi$ ,  $\hat{z}$  are constant values. Therefore, the dual problem of (16) becomes

$$\max_{\beta \in \mathbb{R}} \max_{p \in \mathbb{R}^{n_2}} \min_{z \in \mathbb{R}^{m_2}_+} L\left(z, p, \beta, \hat{z}, \xi, \omega\right).$$
(18)

By solving the inner minimization of the problem (18), duality of the problem (16) is obtained:

$$\max_{\beta \in \mathbb{R}} \max_{p \in \mathbb{R}^{n_2}} \widehat{L}\left(p, \beta, \widehat{z}, \xi\right), \tag{19}$$

where duality function is

$$\widehat{L}(p,\beta,\widehat{z},\xi) = q^{T}(\omega) p - \frac{1}{2} \left\| \left( \widehat{z} + W^{T}(\omega) p + \beta \xi \right)_{+} \right\|^{2} + \beta \widehat{Q}(\xi) + \frac{1}{2} \|\widehat{z}\|^{2}.$$
(20)

The following theorem states that if  $\beta$  is sufficiently large, solving the inner maximization of (19) gives the solution of the problem (16).

**Theorem 3** (see [17]). Consider the following maximization problem

$$\max_{p \in \mathbb{R}^{n_2}} S\left(p, \beta, \hat{z}, \xi, \omega\right) \tag{21}$$

*in which*  $\beta$ ,  $\hat{z}$ , *and*  $\xi$  *are constants, and function*  $S(p, \beta, \hat{z}, \xi)$  *is introduced as follows:* 

$$S(p,\beta,\widehat{z},\xi,\omega) = q^{T}(\omega) p - \frac{1}{2} \left\| \left( \widehat{z} + W^{T}(\omega) p + \beta \xi \right)_{+} \right\|^{2}.$$
(22)

Also, assume that the set  $Z_*$  is nonempty, and the rank of submatrix  $W_l$  of W corresponding to nonzero components of  $\hat{z}_*$  is  $n_2$ . In such a case, there is  $\beta^*$  which for all  $\beta \ge \beta^*$ ,  $\hat{z}_* = (\hat{z} + W^T p(\beta) + \beta\xi)_+$  is the unique and exact solution for the problem (16), where  $p(\beta)$  is the point obtained from solving the problem (21).

Also, in special conditions, the solution for the problem (3) can be also obtained and the following theorem expresses this issue.

**Theorem 4** (see [17]). Assume that the solution set  $Z_*$  is nonempty. For each  $\beta > 0$  and  $\hat{z} \in Z_*$ ,  $y_* = p(\beta)/\beta$  is one exact solution for the linear programming problem (3), where  $p(\beta)$  is the solution for the problem (21).

According to the theorems mentioned above, augmented Lagrangian method presents the following iteration process for solving the problem (16):

$$p_{k+1} \in \arg \max_{p \in \mathbb{R}^{n_2}} \left\{ q^T(\omega) p - \frac{1}{2} \left\| \left( z_k + W(\omega)^T p + \beta \xi \right)_+ \right\|^2 \right\},$$
$$z_{k+1} = \left( z_k + W^T(\omega) p_{k+1} + \beta \xi \right)_+,$$
(23)

where  $z_0$  is an arbitrary vector and here we can use zero vector as initial vector for obtaining normal solution of the problem (4).

We note that the problem (23) is a concave problem and its objective function is piecewise quadratic and is not twice differentiable. Applying the smoothing techniques [18, 19] and replacing  $x_+$  by a smooth approximation, we transform this problem to a twice continuously differentiable problem.

Chen and Mangasarian [19] introduced a family of smoothing functions, which is built as follows. Let  $\rho : R \rightarrow [0, \infty)$  be a piecewise continuous density function satisfying

$$\int_{-\infty}^{+\infty} \rho(s) \, ds = 1, \qquad \int_{-\infty}^{+\infty} |s| \, \rho(s) \, ds < \infty.$$
 (24)

It is obvious that the derivative of plus function is step function, that is,  $(x)_{+} = \int_{-\infty}^{x} \delta(t) dt$ , where the step function  $\delta(x)$  is defined 1 if x > 0 and equals 0 if  $x \le 0$ . Therefore, a smoothing approximation function of the plus function is defined by

$$\varphi(x,\alpha) = \int_{-\infty}^{x} \psi(t,\alpha) dt, \qquad (25)$$

where  $\psi(x, \alpha)$  is smoothing approximation function of step function and is defined as

$$\psi(x,\alpha) = \int_{-\infty}^{x} \alpha \rho(\alpha t) dt.$$
 (26)

By choosing

$$\rho(s) = \frac{e^{-s}}{\left(1 + e^{-s}\right)^2},$$
(27)

specific cases of these approaches are obtained as follows:

$$\psi(x,\alpha) = \frac{1}{1+e^{-\alpha x}} \approx \delta(x),$$

$$\varphi(x,\alpha) = x + \frac{1}{\alpha} \log(1+e^{-\alpha x}) \approx (x)_{+}.$$
(28)

The function  $\varphi$  with a smoothing parameter  $\alpha$  is used here to replace the plus function of (22) to obtain a smooth reformulation of function (22):

$$\widehat{S}(p,\beta,\widehat{z},\xi,\beta,\omega,\alpha) := q^{T}(\omega) p - \left\|\varphi(\widehat{z} + W^{T}(\omega)p + \beta\xi,\alpha)\right\|^{2}.$$
(29)

Therefore, we have the following iterative process instead of (23) and (28):

$$p_{k+1} \in \arg \max_{p \in \mathbb{R}^{n_2}} \left\{ q^T(\omega) p - \left\| \varphi(z_k + W^T(\omega)p + \beta\xi, \alpha) \right\|^2 \right\},$$
$$z_{k+1} = \varphi\left( z_k + W^T(\omega) p_{k+1} + \beta\xi, \alpha \right).$$
(30)

It can be shown that as the smoothing parameter  $\alpha$  approaches infinity any solution of smooth problem (29) approaches the solution of the equivalent problem (22) (see [19]).

We begin with a simple lemma that bounds the square difference between the plus function  $x_+$  and its smooth approximation  $\varphi(x, \alpha)$ .

**Lemma 5** (see [13]). For  $x \in \mathbb{R}$  and  $|x| < \omega$ 

$$\varphi^{2}(x,\alpha) - (x_{+})^{2} \leq \left(\frac{\log(2)}{\alpha}\right)^{2} + \frac{2\overline{\omega}}{\alpha}\log(2),$$
 (31)

where  $\varphi(x, \alpha)$  is the  $\varphi$  function of (28) with smoothing parameter  $\alpha > 0$ .

Theorem 6. Consider the problems (21) and

$$\max_{p \in \mathbb{R}^{n_2}} \widehat{S}\left(p, \beta, \widehat{z}, \xi, \omega, \alpha\right).$$
(32)

*Then, for any*  $p \in \mathbb{R}^{n_2}$  *and*  $\alpha > 0$ 

$$\left| S\left(p,\beta,\widehat{z},\xi,\omega\right) - \widehat{S}\left(p,\beta,\widehat{z},\xi,\omega,\alpha\right) \right| \leq \left(\frac{\log\left(2\right)}{\alpha}\right)^{2} + 2Mm_{2}\frac{\log\left(2\right)}{\alpha},$$
(33)

where M is defined as follows:

$$M = \max_{1 \le i \le m_2} \left| \widehat{z}_i + W_i^T(\omega) p + \beta \xi_i \right|.$$
(34)

Let  $p^*$  be a solution of (21) and  $p^*_{\alpha}$  a solution of (32). Then

$$\max \left\{ S\left(p^{*}, \beta, \widehat{z}, \xi, \omega\right) - S\left(p^{*}_{\alpha}, \beta, \widehat{z}, \xi, \omega\right), \\ \widehat{S}\left(p^{*}_{\alpha}, \beta, \widehat{z}, \xi, \omega, \alpha\right) - \widehat{S}\left(p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right) \right\}$$

$$\leq \left(\frac{\log\left(2\right)}{\alpha}\right)^{2} + 2Mm_{2}\frac{\log\left(2\right)}{\alpha}.$$
(35)

Further, one assumes that  $W^T$  is a full rank matrix. Then,

$$\|p^* - p^*_{\alpha}\| \le \mu \left(\frac{2\log(2)}{\alpha}\right)^2 + 8Mm_2\mu \frac{\log(2)}{\alpha}.$$
 (36)

*Proof.* For any  $\alpha > 0$  and  $p \in \mathbb{R}^{n_2}$ 

$$\varphi\left(\widehat{z}_{i}+W_{i}^{T}\left(\omega\right)p+\beta\xi_{i},\alpha\right)\geq\left(\widehat{z}_{i}+W_{i}^{T}\left(\omega\right)p+\beta\xi_{i},\alpha\right)_{+}.$$
(37)

Hence

$$\begin{aligned} \left| S\left(p,\beta,\widehat{z},\xi,\omega\right) - \widehat{S}\left(p,\beta,\widehat{z},\xi,\omega,\alpha\right) \right| \\ &= S\left(p,\beta,\widehat{z},\xi,\omega\right) - \widehat{S}\left(p,\beta,\widehat{z},\xi,\omega,\alpha\right) \\ &= \left\| \varphi(\widehat{z} + W^{T}(\omega)p + \beta\xi,\alpha) \right\|^{2} \\ &- \left\| \left(\widehat{z} + W^{T}(\omega)p + \beta\xi,\alpha\right)_{+} \right\|^{2} \end{aligned}$$
(38)  
$$&= \sum_{i=1}^{m_{2}} \left( \varphi^{2}\left(\widehat{z}_{i} + W_{i}^{T}(\omega)p + \beta\xi_{i},\alpha\right) \\ &- \left(\widehat{z}_{i} + W_{i}^{T}(\omega)p + \beta\xi_{i},\alpha\right)_{+}^{2} \right). \end{aligned}$$

By using Lemma 5, we get that

$$\begin{split} \left| S\left(p,\beta,\widehat{z},\xi,\omega\right) - \widehat{S}\left(p,\beta,\widehat{z},\xi,\omega,\alpha\right) \right| \\ &\leq \sum_{i=1}^{m_2} \left( \left(\frac{\log\left(2\right)}{\alpha}\right)^2 + 2\left|\widehat{z}_i + W_i^T\left(\omega\right)p + \beta\xi_i,\alpha\right| \frac{\log\left(2\right)}{\alpha} \right) \right. \\ &\leq \left(\frac{\log\left(2\right)}{\alpha}\right)^2 + 2\frac{\log\left(2\right)}{\alpha} \\ &\qquad \times \sum_{i=1}^{m_2} \max_{1 \le i \le m_2} \left|\widehat{z}_i + W_i^T\left(\omega\right)p + \beta\xi_i\right| \\ &= \left(\frac{\log\left(2\right)}{\alpha}\right)^2 + 2Mm_2\frac{\log\left(2\right)}{\alpha}. \end{split}$$
(39)

From above inequality, we have

$$\widehat{S}(p,\beta,\widehat{z},\xi,\omega,\alpha) \leq S(p,\beta,\widehat{z},\xi,\omega)$$

$$\leq \widehat{S}(p,\beta,\widehat{z},\xi,\omega,\alpha) + \left(\frac{\log(2)}{\alpha}\right)^{2}$$

$$+ 2Mm_{2}\frac{\log(2)}{\alpha}.$$
(40)

 TABLE 1: Comparative between smooth augmented Lagrangian Newton method (SALN) and CPLEX solver.

N. P	Recourse problem $n_2 \times m_2 \times d$	Solver	$\ Wz - q\ _{\infty}$	$ \widehat{Q}(\xi) - \xi^T z $	<i>z</i>	Time (second)
D1	50 50 0 60	SALN	9.9135 <i>e</i> - 011	7.2760 <i>e</i> - 012	41.0362	0.3292
F I	$50 \times 50 \times 0.68$	CPLEX	1.0717e - 007	5.2589e - 007	41.0362	0.3182
De		SALN	1.8622 <i>e</i> – 010	4.3074e - 009	57.1793	0.1275
P2	$100 \times 105 \times 0.4$	CPLEX	6.5591 <i>e</i> – 008	1.6105e – 006	57.1826	0.1585
		SALN	3.1559e - 010	5.6361 <i>e</i> - 009	74.9098	0.6593
P3	$150 \times 150 \times 0.5$	CPLEX	5.9572e - 010	1.2014e - 0.09	74 9098	0 1605
		SALN	53819e - 010	1.0506e - 0.08	85 6646	0.2530
P4	$200 \times 200 \times 0.5$	CPLEX	11734e - 007	1.3964e - 0.06	85 6646	0.1820
		SALN	9.4178e = 010	2.7951e - 0.08	102 4325	2 1356
P5	$300 \times 300 \times 0.5$	CPLEX	4.1638e - 010	4.4456e = 0.09	102.4325	0.1830
		SALN	1.2787e = 0.09	2.6226a = 0.08	110 / 189	3 2102
P6	$350 \times 350 \times 0.5$	CDLEY	7.7398a = 0.10	2.0220e = 000	110.4189	0.2116
		CPLEA	7.7398e - 010	5.4452e - 009	124 1204	0.2110
P7	$450\times500\times0.05$	SALN	1.0504e - 010	4.7094e - 009	124.1204	0.7807
		CPLEX	9.0949 <i>e</i> - 013	1.63/1e - 011	124.1205	0.2606
P8	$500 \times 550 \times 0.04$	SALN	1.0425e - 010	1.0241e - 009	134.3999	0.7567
		CPLEX	6.1618 <i>e</i> – 011	1.9081e - 009	134.3999	0.2660
P9	$700 \times 800 \times 0.6$	SALN	4.6139e - 009	2.3908e - 007	153.9782	7.1364
		CPLEX	9.1022e - 009	1.3768e - 007	153.9906	0.8435
P10	$900 \times 1100 \times 0.4$	SALN	5.3396 <i>e</i> – 009	1.5207e - 007	178.1151	7.9383
		CPLEX	4.5020e - 011	2.0373e - 010	178.1163	1.3643
P11 1500 × 2	$1500 \times 2000 \times 0.1$	SALN	1.1289e - 008	2.0696e - 007	231.5284	13.2493
		CPLEX	7.5886 <i>e</i> – 011	1.1059e - 009	231.5286	2.1343
P12 1000 × 2000	$1000 \times 2000 \times 0.01$	SALN	3.6398e - 010	7.9162e - 009	198.4905	5.2620
	1000 × 2000 × 0.01	CPLEX	6.8212e - 013	1.1642e - 010	198.4922	0.6752
P13	$1000 \times 5000 \times 0.001$	SALN	2.7853e - 010	1.9281e - 010	190.0141	0.2709
1 15	1000 × 5000 × 0.001	CPLEX	4.8203e - 010	1.5425e - 009	190.0142	0.2162
D14 1000 × 10000	$1000 \times 10000 \times 0.001$	SALN	1.1221e - 010	1.4988e - 009	212.2416	0.4867
1 14	1000 × 10000 × 0.001	CPLEX	7.9094e - 009	2.6691 <i>e</i> - 007	212.3453	0.3353
D15	1000 1 . 5 0 001	SALN	9.7702 <i>e</i> - 010	2.3283e - 009	231.7930	3.7511
P13	$1000 \times 165 \times 0.001$	CPLEX	2.7285 <i>e</i> - 012	1.8044e - 009	231.8763	1.2472
D16	1000 1 6 0 0000	SALN	9.9432 <i>e</i> - 013	2.0464e - 012	121.3937	9.0948
P10	$1000 \times 166 \times 0.0002$	CPLEX	9.9098 <i>e</i> - 006	1.1089e - 004	121.3940	3.7848
D17	100 1 ( 0.001	SALN	3.9563 <i>e</i> - 010	5.6607 <i>e</i> - 009	73.8493	6.8537
PI/	$100 \times 1e6 \times 0.001$	CPLEX	2.0082e - 003	1.1777 <i>e</i> – 002	73.9412	2.5582
		SALN	2.2737 <i>e</i> – 013	7.9581 <i>e</i> – 013	19.5735	0.0166
P18	$10 \times 1e4 \times 0.001$	CPLEX	2.2737 <i>e</i> – 013	1.1369 <i>e</i> – 013	19.5739	0.1386
		SALN	1.2478e - 0.09	1.0710e - 0.08	18.8192	5.6399
P19	$10 \times 1e6 \times 0.001$	CPLEX	5.9615e - 0.04	64811e - 005	18 8863	2 5623
		SALN	2.0425e - 0.08	1.1816e - 0.08	20.8470	28 7339
P20	$10 \times 5e6 \times 0.01$	CPLEX	4.9966e + 004	$3.3500e \pm 0.05$	0.0000	1 9482
		SALN	3.8654a = 0.12	7.7875e - 012	42.0698	7 8895
P21	$100 \times 1e6 \times 0.01$	CDIEV	1.7094a = 004	7.7875e = 012	42.0098	9.8324
			1.7774t = 0.04	0.2712e = 003	42.0731	0.0324
P22	$100 \times 1e4 \times 0.1$	SALN CDLEV	1.1084e - 012	2.1000e - 012	37.4303	0.1440
		CPLEA	1.2318e - 003	1.31/4e = 003	37.4303	0.3099
P23	$100\times 1e5\times 0.05$	SALN	1.70556 - 012	0.3121e - 012	45.5944	1.13/9
		CPLEX	3.982/e - 006	2.8309 <i>e</i> - 006	45.5944	1.5729

TABLE 1: Continued.

N. P	Recourse problem $n_2 \times m_2 \times d$	Solver	$\left\ Wz-q\right\ _{\infty}$	$ \widehat{Q}(\xi) - \xi^T z $	<i>z</i>	Time (second)
P24	$1000 \times 5 24 \times 0.1$	SALN	1.3074e - 012	2.4102 <i>e</i> - 011	117.4693	15.1065
	$1000 \times 3e4 \times 0.1$	CPLEX	6.7455e - 007	6.7379 <i>e</i> – 006	117.4699	20.3967
P25	1000 × 1.5 × 0.09	SALN	2.5011 <i>e</i> - 012	1.0516e – 011	116.6964	23.2540
	1000 × 163 × 0.08	CPLEX	9.4298e - 006	5.5861e - 004	116.6964	33.4319

Therefore

$$S(p^{*}, \beta, \widehat{z}, \xi, \omega) - S(p^{*}_{\alpha}, \beta, \widehat{z}, \xi, \omega)$$

$$\leq \widehat{S}(p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha) - \widehat{S}(p^{*}_{\alpha}, \beta, \widehat{z}, \xi, \omega, \alpha)$$

$$+ \left(\frac{\log(2)}{\alpha}\right)^{2} + 2Mm_{2}\frac{\log(2)}{\alpha}$$

$$\leq \left(\frac{\log(2)}{\alpha}\right)^{2} + 2Mm_{2}\frac{\log(2)}{\alpha},$$

$$\widehat{S}(p^{*}_{\alpha}, \beta, \widehat{z}, \xi, \omega, \alpha) - \widehat{S}(p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha)$$

$$\leq S(p^{*}_{\alpha}, \beta, \widehat{z}, \xi, \omega) - S(p^{*}, \beta, \widehat{z}, \xi, \omega)$$

$$+ \left(\frac{\log(2)}{\alpha}\right)^{2} + 2Mm_{2}\frac{\log(2)}{\alpha}$$

$$\leq \left(\frac{\log(2)}{\alpha}\right)^{2} + 2Mm_{2}\frac{\log(2)}{\alpha}.$$
(41)

Suppose that  $W^T$  is full rank. Then the Hessian of  $\hat{S}$  is negative definite, and  $\hat{S}$  is strongly concave on bounded sets. By the definition of strong concavity, for any  $\gamma \in (0, 1)$ ,

$$\widehat{S}(\gamma p_{\alpha}^{*} + (1 - \gamma) p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha) - \gamma \widehat{S}(p_{\alpha}^{*}, \beta, \widehat{z}, \xi, \omega, \alpha)$$
$$- (1 - \gamma) \widehat{S}(p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha)$$
$$\geq \frac{1}{2} \mu \gamma (1 - \gamma) \| p_{\alpha}^{*} - p^{*} \|^{2}.$$
(42)

Let  $\gamma = 1/2$ , then

$$\begin{aligned} \frac{1}{8}\mu \|p_{\alpha}^{*} - p^{*}\|^{2} \\ &\leq \widehat{S}\left(\frac{1}{2}\left(p_{\alpha}^{*} + p^{*}\right), \beta, \widehat{z}, \xi, \omega, \alpha\right) \\ &- \frac{1}{2}\left(\widehat{S}\left(p_{\alpha}^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right) + \widehat{S}\left(p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right)\right) \\ &\leq \widehat{S}\left(p_{\alpha}^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right) \\ &- \frac{1}{2}\left(\widehat{S}\left(p_{\alpha}^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right) + \widehat{S}\left(p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right)\right) \\ &\leq \frac{1}{2}\left(\widehat{S}\left(p_{\alpha}^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right) - \widehat{S}\left(p^{*}, \beta, \widehat{z}, \xi, \omega, \alpha\right)\right) \\ &\leq \frac{1}{2}\left(\frac{\log\left(2\right)}{\alpha}\right)^{2} + Mm_{2}\frac{\log\left(2\right)}{\alpha}. \end{aligned}$$

43)

Considering the advantage of the twice differentiability of the objective function of the problem (32) allows us to use a quadratically convergent Newton algorithm with an Armijo stepsize [20] that makes the algorithm globally convergent.

#### 4. Numerical Results and Algorithm

In each iteration of the process (30), one concave, quadratic, unconstrained maximization problem is solved. For solving it, the fast Newton method can be used.

In the algorithm, the Hessian matrix may be singular, thus we use a modified Newton. The direction in each iteration for solving (30) is obtained through the following relation:

$$d_{s} = -\left(\nabla_{p}^{2}\widehat{S}\left(p,\beta,\widehat{z},\xi,\omega,\alpha\right) - \delta I_{n_{2}}\right)^{-1}\left(\nabla_{p}\widehat{S}\left(p,\beta,\widehat{z},\xi,\omega,\alpha\right)\right),$$
$$p_{s+1} = p_{s} + \lambda_{s}d_{s},$$
(44)

where  $\delta$  is a small positive number,  $I_{n_2}$  is the identity matrix of order  $n_2$ , and  $\lambda_s$  is the suitable step length that Armijo algorithm is used for determining it (see Algorithm 1).

The proposed algorithm was applied to solve some recourse problems. Table 1 compares this algorithm with CPLEX v. 12.1 solver for quadratic convex programming problems (5). As is evident from Table 1, most of recourse problems could be solved more successful by the algorithm which is based on smooth augmented Lagrangian Newton method (SALN) than CPLEX package (for illustration see the problems 21–25 in Table 1). This algorithm gives us high accuracy and the solution with minimum norm in suitable time (see last column of Table 1). Also, we can find that CPLEX is better than the algorithm proposed for some recourse problems in which the matrices are approximately square (Ex. line 5–12).

The test generator generates recourse problems. These problems are generated using the MATLAB code show in Algorithm 2.

The algorithm considered for solving several recourse problems was run on a computer with 2.5 dual-core CPU and 4 GB memory in MATLAB 7.8 programming environment. Also, in the generated problems, recourse matrix W is the Sparse matrix  $(n_2 \times m_2)$  with the density d. The constants  $\beta$  and  $\delta$  in the above algorithm in (44) were selected 1 and 10<sup>-8</sup>, respectively.

In Table 1, the second column indicates the size and density of matrix W, the forth column indicates the feasibility of the primal problem (4), and the next column indicates the error norm function of this problem (the MATLAB code of this paper is available from the authors upon request).

```
Choose a z_0 \in \mathbb{R}^{m_2}, \alpha > 1, i > 1, \epsilon > 0 be error tolerance and \delta is a small positive number.

i = 0;

While ||z_i - z_{i-1}||_{\infty} \ge \epsilon

Choose a p_0 \in \mathbb{R}^{n_2} and set k = 0.

While ||\nabla_p \widehat{S}(p_k, \beta, z_i, \xi, \omega, \alpha)||_{\infty} \ge \epsilon

Choose \lambda_k = \max\{s, s\sigma, s\sigma^2, \ldots\} such that

\widehat{S}(p_k, \beta, z_i, \xi, \omega, \alpha) - \widehat{S}(p_k + \lambda_k d_k, \beta, z_i, \xi, \omega, \alpha) \ge -\lambda_i \mu \nabla (\widehat{S}(p_k, \beta, z_i, \xi, \omega, \alpha))^T d_k,

where,

d_k = -(\nabla^2 \widehat{S}(p_k, \beta, z_i, \xi, \omega, \alpha, ) - \delta I_{n_2})^{-1} \nabla \widehat{S}(p_k, \beta, z_i, \xi, \omega, \alpha), s > 0 be a constant,

\sigma \in (0, 1) and \mu \in (0, 1).

Put p_{k+1} = p_k + \lambda_k d_k, k = k + 1 and \alpha = i\alpha.

end

Set p_{i+1} = p_{k+1}, z_{i+1} = \varphi(z_i + W^T(\omega)p_{i+1} + \beta\xi, \alpha) and i = i + 1.

end
```

ALGORITHM 1: Newton method with the Armijo rule.

```
\label{eq:spectral_series} \begin{split} & \mbox{``Sgen: Generate random solvable recourse problems:} \\ & \mbox{``Input: m,n,d(ensity); Output: W,q,\xi;} \\ & \mbox{``m=input(`Enter $n_2$:')} \\ & \mbox{``n=input(`Enter $m_2$:')} \\ & \mbox{``d=input(`Enter $d:')} \\ & \mbox{``pl=inline('(abs(x)+x)/2')} \\ & \mbox{``m=sprand($n_2$,$m_2$,$d);W=100*(W=0.5*spones(W));} \\ & \mbox{``ssparse(10*pl(rand($m_2$,1)));} \\ & \mbox{``ssparse(10*pl(rand($m_2$,1)));} \\ & \mbox{``ssparse(isign(pl(rand($n_2$,1)-rand($n_2$,1)))),0,$n_2$,$n_2$)} \\ & \mbox{``ssparse((rand($m_2$,1)-rand($n_2$,1)));} \\ & \mbox{``ssparse(isign($pl(rand($m_2$,1)-rand($n_2$,1)))),0,$m_2$,$n_2$)} \\ & \mbox{``ssparse($m_2$,1)-rand($n_2$,1));} \\ & \mbox{``ssparse($m_2$,1)-sign($z)$),0,$m_2$,$m_2$)*ones($m_2$,1));} \\ & \mbox{``ssparse($m_2$,1)-rign($z)$),0,$m_2$,$m_2$)*ones($m_2$,1));} \\ & \mbox{``ssparse($m_2$,1)-rign($z)$),0,$m_2$,$m_2$)*ones($m_2$,1),0;} \\ & \mbox{``ssparse($m_2$,1)-rign($z)$),0,$m_2$,m_2$)*ones($m_2$,1),0;} \\ & \mbox{``ssparse($m_2$,1)-rign($m_2$,1),0;} \\ & \mbox{
```



#### 5. Conclusion

In this paper, a smooth reformulation process, based on augmented Lagrangian algorithm, was proposed for obtaining the normal solution of recourse problem of a stochastic linear programming. This smooth iterative process allows us to use a quadratically convergent Newton algorithm, which accelerates obtaining the normal solution.

Table 1 shows that the proposed algorithm has appropriate speed in most of the problems. This result, specifically, can be observed in recourse problems with the matrix of coefficients in which the number of constraints is noticeably more than the number of variables. The more challenging is solving the problems which their coefficient matrix is square (the numbers of constraints and variables get closer to each other), and more time is needed by the algorithm for solving the problem.

#### Acknowledgment

The authors would like to thank the reviewers for their helpful comments.

#### References

- J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer Series in Operations Research, Springer, New York, NY, USA, 1997.
- [2] G. B. Dantzig, "Linear programming under uncertainty," Management Science. Journal of the Institute of Management Science. Application and Theory Series, vol. 1, pp. 197–206, 1955.
- [3] J. L. Higle and S. Sen, "Stochastic decomposition: an algorithm for two-stage linear programs with recourse," *Mathematics of Operations Research*, vol. 16, no. 3, pp. 650–669, 1991.
- [4] P. Kall and S. W. Wallace, Stochastic Programming, John Wiley & Sons, Chichester, UK, 1994.
- [5] S. A. Tarim and I. A. Miguel, "A hybrid benders' decomposition method for solving stochastic constraint programs with linear recourse," in *Recent Advances in Constraints*, vol. 3978 of *Lecture Notes in Computer Science*, pp. 133–148, Springer, Berlin, Germany, 2006.
- [6] J. R. Birge, X. Chen, L. Qi, and Z. Wei, "A stochastic Newton method for stochastic quadratic programs with recourse," Tech. Rep., Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, Mich, USA, 1994.

- [7] X. Chen, "A parallel BFGS-SQP method for stochastic linear programs," in *World Scientific*, pp. 67–74, World Scientific, River Edge, NJ, USA, 1995.
- [8] X. Chen, "Newton-type methods for stochastic programming," Mathematical and Computer Modelling, vol. 31, no. 10–12, pp. 89–98, 2000.
- [9] X. J. Chen, L. Q. Qi, and R. S. Womersley, "Newton's method for quadratic stochastic programs with recourse," *Journal of Computational and Applied Mathematics*, vol. 60, no. 1-2, pp. 29–46, 1995.
- [10] X. Chen and R. S. Womersley, "A parallel inexact Newton method for stochastic programs with recourse," *Annals of Operations Research*, vol. 64, pp. 113–141, 1996.
- [11] X. Chen and R. S. Womersley, "Random test problems and parallel methods for quadratic programs and quadratic stochastic programs," *Optimization Methods and Software*, vol. 13, no. 4, pp. 275–306, 2000.
- [12] X. Chen and Y. Ye, "On homotopy-smoothing methods for boxconstrained variational inequalities," *SIAM Journal on Control and Optimization*, vol. 37, no. 2, pp. 589–616, 1999.
- [13] Y.-J. Lee and O. L. Mangasarian, "SSVM: a smooth support vector machine for classification," *Computational Optimization* and Applications. An International Journal, vol. 20, no. 1, pp. 5– 22, 2001.
- [14] C. Kanzow, H. Qi, and L. Qi, "On the minimum norm solution of linear programs," *Journal of Optimization Theory and Applications*, vol. 116, no. 2, pp. 333–345, 2003.
- [15] O. L. Mangasarian, "A Newton method for linear programming," *Journal of Optimization Theory and Applications*, vol. 121, no. 1, pp. 1–18, 2004.
- [16] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, USA, 1970.
- [17] Yu. G. Evtushenko, A. I. Golikov, and N. Mollaverdy, "Augmented Lagrangian method for large-scale linear programming problems," *Optimization Methods & Software*, vol. 20, no. 4-5, pp. 515–524, 2005.
- [18] C. H. Chen and O. L. Mangasarian, "Smoothing methods for convex inequalities and linear complementarity problems," *Mathematical Programming*, vol. 71, no. 1, pp. 51–69, 1995.
- [19] C. Chen and O. L. Mangasarian, "A class of smoothing functions for nonlinear and mixed complementarity problems," *Computational Optimization and Applications. An International Journal*, vol. 5, no. 2, pp. 97–138, 1996.
- [20] X. Chen, L. Qi, and D. Sun, "Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities," *Mathematics* of Computation, vol. 67, no. 222, pp. 519–540, 1998.

# Research Article

# New Results on Robust Stability and Stabilization of Linear Discrete-Time Stochastic Systems with Convex Polytopic Uncertainties

## P. Niamsup<sup>1,2</sup> and G. Rajchakit<sup>3</sup>

<sup>1</sup> Department of Mathematics, Faculty of Science, Chiang Mai University, Chiangmai 50200, Thailand

<sup>2</sup> Center of Excellence in Mathematics, CHE, Si Ayutthaya Road, Bangkok 10400, Thailand

<sup>3</sup> Division of Mathematics and Statistics, Faculty of Science, Maejo University, Chiangmai 50290, Thailand

Correspondence should be addressed to G. Rajchakit; mrajchakit@yahoo.com

Received 17 January 2013; Revised 1 April 2013; Accepted 15 April 2013

Academic Editor: Rung Ching Chen

Copyright © 2013 P. Niamsup and G. Rajchakit. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper addresses the robust stability for a class of linear discrete-time stochastic systems with convex polytopic uncertainties. The system to be considered is subject to both interval time-varying delays and convex polytopic type uncertainties. Based on the augmented parameter-dependent Lyapunov-Krasovskii functional, new delay-dependent conditions for the robust stability are established in terms of linear matrix inequalities. An application to robust stabilization of linear discrete-time stochastic control systems is given. Numerical examples are included to illustrate the effectiveness of our results.

#### 1. Introduction

In the past decades, the problem of stability for neutral differential systems, which have delays in both its state and the derivatives of its states, has been widely investigated by many researchers. Such systems are often encountered in engineering, biology, and economics. The existence of time delay is frequently a source of instability or poor performances in the systems. Recently, some stability criteria for neutral system with time delay have been given in [1-8] and the references therein. Some delay-dependent stability criteria for discrete-time systems with time-varying delay are investigated in [2, 6, 9–11], where the discrete Lyapunov functional method is employed to prove stability conditions in terms of linear matrix inequalities (LMIs). A number of research works for dealing with asymptotic stability problem for discrete systems with interval time-varying delays have been presented in [12-24]. Theoretically, stability analysis of the systems with time-varying delays is more complicated, especially for the case where the system matrices belong to some convex polytope. In this case, the parameter-dependent Lyapunov-Krasovskii functionals are constructed as the convex combination of a set of functions assures the robust

stability of the nominal systems, and the stability conditions must be solved upon a grid on the parameter space, which results in testing a finite number of linear matrix inequalities (LMIs) [11, 25, 26]. To the best of the authors' knowledge, the stability for linear discrete-time systems with both timevarying delays and polytopic uncertainties has not been fully investigated. The papers [27, 28] propose sufficient conditions for robust stability of discrete and continuous polytopic systems without time delays. More recently, combining the ideas in [25, 26], improved conditions for D-stability and D-stabilization of linear polytopic delay-difference equations with constant delays have been proposed in [29]. To the best of our knowledge, the stability and stabilization of linear discrete-time stochastic systems with convex polytopic uncertainties, nondifferentiable time-varying delays has not been fully studied yet (see, e.g., [1, 3-11, 13-36] and the references therein), which are important in both theories and applications. This motivates our research.

In this paper, we consider polytopic discrete-time stochastic equations with interval time-varying delays. By using the parameter-dependent Lyapunov-Krasovskii functional combined with LMI techniques, we propose new criteria for the robust stability of the stochastic system. The
delay-dependent stability conditions are formulated in terms of LMIs, being thus solvable by the numeric technology available in the literature to date. The result is applied to robust stabilization of linear discrete-time stochastic control systems. Compared to other results, our result has its own advantages. First, it deals with the delay-difference stochastic system, where the state-space data belong to the convex polytope of uncertainties and the rate of change of the state depends not only on the current state of the systems, but also its state at some times in the past. Second, the time delay is assumed to be a time-varying function belonging to a given interval, which means that the lower and upper bounds for the time-varying delay are available. Third, our approach allows us to apply in robust stabilization of the linear discretetime stochastic system subjected to polytopic uncertainties and external controls. Therefore, our results are more general than the related previous results.

The paper is organized as follows. Section 2 introduces the main notations, definitions, and some lemmas needed for the development of the main results. In Section 3, sufficient conditions are derived for robust stability, stabilization of discrete-time stochastic systems with interval time-varying delays, and polytopic uncertainties. They are followed by some remarks. Illustrative examples are given in Section 4.

### 2. Preliminaries

The following notations will be used throughout this paper.  $R^+$  denotes the set of all real nonnegative numbers;  $R^n$  denotes the *n*-dimensional space with the scalar product  $\langle \cdot, \cdot \rangle$  and the vector norm  $\|\cdot\|$ ;  $R^{n\times r}$  denotes the space of all real matrices of  $(n \times r)$ -dimension.  $A^T$  denotes the transpose of A; a matrix A is symmetric if  $A = A^T$ , and a matrix I is the identity matrix of appropriate dimension.

Matrix *A* is semipositive definite  $(A \ge 0)$  if  $\langle Ax, x \rangle \ge 0$ , for all  $x \in \mathbb{R}^n$ ; *A* is positive definite (A > 0) if  $\langle Ax, x \rangle > 0$  for all  $x \ne 0$ ;  $A \ge B$  means that  $A - B \ge 0$ .

Consider delay-difference stochastic systems with polytopic uncertainties of the form

$$x (k + 1) = A (\xi) x (k) + D (\xi) x (k - h (k)) + \sigma (x (k), x (k - h (k)), k) \omega (k), k = 0, 1, 2, ..., (1) x (k) = v_k, \quad k = -h_2, -h_2 + 1, ..., 0,$$

where  $x(k) \in \mathbb{R}^n$  is the state (Figures 1 and 2), and the system matrices are subjected to uncertainties and belong to the polytope  $\Omega$  given by

$$\Omega = \left\{ [A, D] (\xi) := \sum_{i=1}^{p} \xi_i [A_i, D_i], \sum_{i=1}^{p} \xi_i = 1, \xi_i \ge 0 \right\}, \quad (2)$$

where  $A_i$ ,  $D_i$ , i = 1, 2, ..., p, are given constant matrices with appropriate dimensions,  $\omega(k)$  is a scalar Wiener process (Brownian Motion) on  $(\Omega, \mathcal{F}, \mathcal{P})$  with

$$E[\omega(k)] = 0, \qquad E[\omega^{2}(k)] = 1,$$
  

$$E[\omega(i)\omega(j)] = 0 \quad (i \neq j),$$
(3)



FIGURE 1: The simulation of the solutions  $x_1(k)$  and  $x_2(k)$  with the initial condition  $\phi(k) = \begin{bmatrix} 10 & 5 \end{bmatrix}^T$ ,  $k \in [0, 10]$ .



FIGURE 2: The simulation of the solutions  $x_1(k)$  and  $x_2(k)$  with the initial condition  $\phi(k) = \begin{bmatrix} 10 & 5 \end{bmatrix}^T$ ,  $k \in [0, 10]$ .

and  $\sigma: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$  is the continuous function and is assumed to satisfy that

$$\sigma^{T}(x(k), x(k - h(k)), k) \sigma(x(k), x(k - h(k)), k)$$

$$\leq \rho_{1}x^{T}(k) x(k) + \rho_{2}x^{T}(k - h(k)) x(k - h(k)), \quad (4)$$

$$x(k), x(k - h(k)) \in \mathbb{R}^{n},$$

where  $\rho_1 > 0$  and  $\rho_2 > 0$  are known constant scalars.

For simplicity, we denote  $\sigma(x(k), x(k - h(k)), k)$  by  $\sigma$ , respectively.

The time-varying function h(k) satisfies the condition:

$$0 < h_1 \le h(k) \le h_2, \quad \forall k = 0, 1, 2, \dots$$
 (5)

*Remark 1.* It is worth noting that the time delay is a timevarying function belonging to a given interval, which allows the time delay to be a fast time-varying function, and the lower bound is not restricted to being zero as considered in [2, 6, 9–11, 18–24, 30–33].

Definition 2. The system (1) is robustly stable in the mean square if there exists a positive definite scalar function  $V(k, x(k)): \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$  such that

$$E [\Delta V (k, x (k))] = E [V (k + 1, x (k + 1)) - V (k, x (k))] < 0,$$
(6)

along any trajectory of zero solution of the system (1) for all uncertainties in  $\Omega$ .

**Proposition 3.** For real numbers  $\xi_i \ge 0$ , i = 1, 2, ..., p,  $\sum_{i=1}^{p} \xi_i = 1$ , the following inequality holds:

$$(p-1)\sum_{i=1}^{p}\xi_{i}^{2}-2\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\xi_{i}\xi_{j}\geq0.$$
(7)

Proof. The proof is followed from completing the square

$$(p-1)\sum_{i=1}^{p}\xi_{i}^{2} - 2\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\xi_{i}\xi_{j} = \sum_{i=1}^{p-1}\sum_{j=i+1}^{p}(\xi_{i} - \xi_{j})^{2} \ge 0.$$
(8)

# 3. Main Results

*3.1. Robust Stability.* In this section, we present sufficient delay-dependent conditions for the robust stability of system (1). Let us set

$$||x_k|| = \sup_{s \in [-h_2, 0]} ||x(k+s)||$$

$$\mathcal{M}_{ij}(P,Q,S_{1},S_{2}) = \begin{pmatrix} (h_{2}-h_{1}+1)Q_{i}-P_{i}-S_{1i}A_{j}-A_{j}^{T}S_{1i}^{T}+2\rho_{1}I & S_{1i}-S_{1i}A_{j} & -S_{1i}D_{j}-S_{2i}A_{j} \\ S_{1i}^{T}-A_{j}^{T}S_{1i}^{T} & P_{i}+S_{1i}+S_{1i}^{T} & S_{2i}-S_{1i}D_{j} \\ -D_{j}^{T}S_{1i}^{T}-A_{j}^{T}S_{2i}^{T} & S_{2i}^{T}-D_{j}^{T}S_{1i}^{T} & -Q_{i}-S_{2i}D_{j}-D_{j}^{T}S_{2i}^{T}+2\rho_{2}I \end{pmatrix},$$
(9)  
$$\mathcal{S} = \begin{pmatrix} S & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad P(\xi) = \sum_{i=1}^{p} \xi_{i}P_{i}, \qquad Q(\xi) = \sum_{i=1}^{p} \xi_{i}Q_{i}, \qquad S_{1}(\xi) = \sum_{i=1}^{p} \xi_{i}S_{1i}, \qquad S_{2}(\xi) = \sum_{i=1}^{p} \xi_{i}S_{2i}.$$

**Theorem 4.** The system (1) is robustly stable in the mean square if there exist symmetric matrices  $P_i > 0$ ,  $Q_i > 0$ , i = 1, 2, ..., p, and constant matrices  $S \ge 0$ ,  $S_{1i}$ ,  $S_{2i}$ , i = 1, 2, ..., p, satisfying the following LMIs:

(i) 
$$\mathcal{M}_{ii}(P,Q,S_1,S_2) + \mathcal{S} < 0, i = 1, 2, ..., p;$$
  
(ii)  $\mathcal{M}_{ij}(P,Q,S_1,S_2) + \mathcal{M}_{ji}(P,Q,S_1,S_2) - (2/(p-1))\mathcal{S} < 0, i = 1, 2, ..., p - 1; j = i + 1, ..., p.$ 

*Proof.* Consider the following parameter-dependent Lyapunov-Krasovskii functional for system (1):

 $V_{1}(k) = x(k) P(\xi) x(k),$ 

$$V(k) = V_1(k) + V_2(k) + V_3(k), \qquad (10)$$

where

$$V_{2}(k) = \sum_{i=k-h(k)}^{k-1} x^{T}(i) Q(\xi) x(i),$$
(11)

$$V_{3}(k) = \sum_{j=-h_{2}+2}^{-h_{1}+1} \sum_{l=k+j+1}^{k-1} x^{T}(l) Q(\xi) x(l).$$

We can verify that

$$\lambda_1 \| x(k) \|^2 \le V(k) \le \lambda_2 \| x_k \|^2.$$
(12)

Let us set  $z(k) = [x^{T}(k) x^{T}(k+1) x^{T}(k-h(k)) \omega^{T}(k)]$ , and

$$G\left(\xi\right) = \begin{pmatrix} 0 & 0 & 0 & 0\\ 0 & P\left(\xi\right) & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad F\left(\xi\right) = \begin{pmatrix} P\left(\xi\right) & 0 & 0 & 0\\ I & I & 0 & 0\\ 0 & 0 & I & 0\\ 0 & 0 & 0 & I \end{pmatrix}.$$
(13)

Then, with the difference of  $V_1(k)$  along the solution of the system (1) and taking the mathematical expectation, we obtained

$$E \left[ \Delta V_{1} \left( k \right) \right]$$
  
=  $E \left[ x^{T} \left( k + 1 \right) P \left( \xi \right) x \left( k + 1 \right) - x^{T} \left( k \right) P \left( \xi \right) x \left( k \right) \right]$   
=  $E \left[ z(k)^{T} G \left( \xi \right) z \left( k \right) - 2z^{T} \left( k \right) F^{T} \left( \xi \right) \begin{pmatrix} 0.5x \left( k \right) \\ 0 \\ 0 \\ 0 \end{pmatrix} \right],$   
(14)

because of

$$z^{T}(k) G(\xi) z(k) = x(k+1)^{T} P(\xi) x(k+1),$$
  

$$2z^{T}(k) F^{T}(\xi) \begin{pmatrix} 0.5x(k) \\ 0 \\ 0 \\ 0 \end{pmatrix} = x^{T}(k) P(\xi) x(k).$$
(15)

$$0 = -S_{2}(\xi) x (k + 1) + S_{2}(\xi) A (\xi) x (k) + S_{2}(\xi) D (\xi) x (k - h (k)) + S_{2}(\xi) \sigma \omega (k), 0 = -\sigma^{T} x (k + 1) + \sigma^{T} A (\xi) x (k) + \sigma^{T} D (\xi) x (k - h (k)) + \sigma^{T} \sigma \omega (k),$$
(16)

Using the expression of system (1)

$$\begin{split} 0 &= -S_1 \left( \xi \right) x \left( k + 1 \right) + S_1 \left( \xi \right) A \left( \xi \right) x \left( k \right) \\ &+ S_1 \left( \xi \right) D \left( \xi \right) x \left( k - h \left( k \right) \right) + S_1 \left( \xi \right) \sigma \omega \left( k \right), \end{split}$$

we have

$$- 2z^{T}(k) F^{T}(\xi) \begin{pmatrix} 0.5x(k) \\ -S_{1}(\xi) x(k+1) + S_{1}(\xi) A(\xi) x(k) + S_{1}(\xi) D(\xi) x(k-h(k)) + S_{1}(\xi) \sigma \omega(k) \\ -S_{2}(\xi) x(k+1) + S_{2}(\xi) A(\xi) x(k) + S_{2}(\xi) D(\xi) x(k-h(k)) + S_{2}(\xi) \sigma \omega(k) \\ -\sigma^{T} x(k+1) + \sigma^{T} A(\xi) x(k) + \sigma^{T} D(\xi) x(k-h(k)) + \sigma^{T} \sigma \omega(k) \end{pmatrix} z(k)$$

$$= -z^{T}(k) F^{T}(\xi) \begin{pmatrix} 0.5I & 0 & 0 \\ S_{1}(\xi) A(\xi) & -S_{1}(\xi) S_{1}(\xi) D(\xi) S_{1}(\xi) \sigma \\ S_{2}(\xi) A(\xi) & -S_{2}(\xi) S_{2}(\xi) D(\xi) S_{2}(\xi) \sigma \\ \sigma^{T} A(\xi) & -\sigma^{T} & \sigma^{T} D(\xi) & \sigma^{T} \sigma \end{pmatrix}^{T} F(\xi) z(k).$$

$$(17)$$

Therefore, from (14), it follows that

$$E \left[ \Delta V_{1} \left( k \right) \right]$$

$$= E \left[ x^{T}(k) \left[ -P\left(\xi\right) - S_{1}\left(\xi\right) A\left(\xi\right) - A\left(\xi\right)^{T} S_{1}^{T}(\xi) \right] x\left(k\right) \right]$$

$$+ 2x^{T}(k) \left[ S_{1}\left(\xi\right) - S_{1}\left(\xi\right) A\left(\xi\right) \right] x\left(k+1\right) + 2x^{T}(k) \left[ -S_{1}\left(\xi\right) D\left(\xi\right) - S_{2}\left(\xi\right) A\left(\xi\right) \right] x\left(k-h\left(k\right)\right) + 2x^{T}(k) \left[ -S_{1}\left(\xi\right) \sigma - \sigma^{T} A\left(\xi\right) \right] \omega\left(k\right) + x\left(k+1\right) \left[ P\left(\xi\right) + S_{1}\left(\xi\right) + S_{1}^{T}\left(\xi\right) \right] \\ \times x\left(k+1\right) + 2x\left(k+1\right) \left[ S_{2}\left(\xi\right) - S_{1}\left(\xi\right) D\left(\xi\right) \right] \\ \times x\left(k+1\right) + 2x\left(k+1\right) \left[ \sigma^{T} - S_{1}\left(\xi\right) \sigma \right] \omega\left(k\right) + x^{T}(k-h\left(k\right)) \left[ -S_{2}\left(\xi\right) D\left(\xi\right) - D^{T}(\xi) S_{2}^{T}(\xi) \right] \\ \times x\left(k-h\left(k\right)\right) + x^{T}(k-h\left(k\right)) \left[ -S_{2}\left(\xi\right) \sigma - \sigma^{T} D\left(\xi\right) \right] \omega\left(k\right) + \omega^{T}(k) \left[ -2\sigma^{T}\sigma \right] \omega\left(k\right) \right].$$
(18)

By assumption (3), we have

$$E\left[\Delta V_{1}(k)\right]$$

$$= E\left[x^{T}(k)\left[-P\left(\xi\right) - S_{1}\left(\xi\right)A\left(\xi\right) - A\left(\xi\right)^{T}S_{1}^{T}(\xi)\right]x\left(k\right) + 2x^{T}(k)\left[S_{1}\left(\xi\right) - S_{1}\left(\xi\right)A\left(\xi\right)\right]x\left(k+1\right) + 2x^{T}(k)\left[-S_{1}\left(\xi\right)D\left(\xi\right) - S_{2}\left(\xi\right)A\left(\xi\right)\right] \times x\left(k-h\left(k\right)\right) + x\left(k+1\right)\left[P\left(\xi\right) + S_{1}\left(\xi\right) + S_{1}^{T}\left(\xi\right)\right]x\left(k+1\right) + 2x\left(k+1\right)\left[S_{2}\left(\xi\right) - S_{1}\left(\xi\right)D\left(\xi\right)\right]x\left(k-h\left(k\right)\right) + x^{T}(k-h\left(k\right))\left[-S_{2}\left(\xi\right)D\left(\xi\right) - D^{T}\left(\xi\right)S_{2}^{T}\left(\xi\right)\right] \times x\left(k-h\left(k\right)\right) + \omega^{T}(k)\left[-2\sigma^{T}\sigma\right]\omega\left(k\right)\right].$$
(19)

Applying assumption (4), the following estimations holds:

$$-\sigma^{T}(x(k), x(k - h(k)), k)\sigma_{i}(x(k), x(k - h(k)), k)$$

$$\leq \rho_{1}x^{T}(k)x(k) + \rho_{2}x^{T}(k - h(k))x(k - h(k)).$$
(20)

Therefore, we have

Е

$$\begin{split} \left[ \Delta V_{1} \left( k \right) \right] \\ &= E \left[ x^{T}(k) \left[ -P\left( \xi \right) - S_{1} \left( \xi \right) A\left( \xi \right) - A\left( \xi \right)^{T} S_{1}^{T}(\xi) \right. \\ &+ 2\rho_{1} I \right] x\left( k \right) \\ &+ 2x^{T}(k) \left[ S_{1} \left( \xi \right) - S_{1} \left( \xi \right) A\left( \xi \right) \right] x\left( k + 1 \right) \\ &+ 2x^{T}(k) \left[ -S_{1} \left( \xi \right) D\left( \xi \right) - S_{2} \left( \xi \right) A\left( \xi \right) \right] \\ &\times x\left( k - h\left( k \right) \right) \\ &+ x\left( k + 1 \right) \left[ P\left( \xi \right) + S_{1} \left( \xi \right) + S_{1}^{T}(\xi) \right] \\ &\times x\left( k + 1 \right) \\ &+ 2x\left( k + 1 \right) \left[ S_{2} \left( \xi \right) - S_{1} \left( \xi \right) D\left( \xi \right) \right] \\ &\times x\left( k - h\left( k \right) \right) \\ &+ x^{T}(k - h\left( k \right)) \left[ -S_{2} \left( \xi \right) D\left( \xi \right) - D^{T} \left( \xi \right) S_{2}^{T}(\xi) \\ &+ 2\rho_{2} I \right] x\left( k - h\left( k \right) \right) \left] . \end{split}$$

$$(21)$$

The expectation of the difference of  $V_2(k)$  is given by

$$E\left[\Delta V_{2}(k)\right]$$

$$= E\left[\sum_{i=k+1-h(k+1)}^{k} x^{T}(i) Q(\xi) x(i) - \sum_{i=k-h(k)}^{k-1} x^{T}(i) Q(\xi) x(i)\right]$$

$$= E\left[\sum_{i=k+1-h(k+1)}^{k-h_{1}} x^{T}(i) Q(\xi) x(i) + x^{T}(k) Q(\xi) x(k) - x^{x}(k-h(k)) Q(\xi) x(k-h(k)) + \sum_{i=k+1-h_{1}}^{k-1} x^{T}(i) Q(\xi) x(i) - \sum_{i=k+1-h(k)}^{k-1} x^{T}(i) Q(\xi) x(i)\right].$$
(22)

Since  $h(k) \ge h_1$ , we have

$$\sum_{i=k+1-h_{1}}^{k-1} x^{T}(i) Q(\xi) x(i) - \sum_{i=k+1-h(k)}^{k-1} x^{T}(i) Q(\xi) x(i) \le 0, \quad (23)$$

and, hence, from (22), we have

$$E\left[\Delta V_{2}(k)\right] \leq E\left[\sum_{i=k+1-h(k+1)}^{k-h_{1}} x^{T}(i) Q(\xi) x(i) + x^{T}(k) Q(\xi) x(k) -x^{T}(k-h(k)) Q(\xi) x(k-h(k))\right].$$
(24)

The difference of  $V_3(k)$  is given by

$$E\left[\Delta V_{3}(k)\right]$$

$$= E\left[\sum_{j=-h_{2}+2}^{-h_{1}+1} \left[x^{T}(k)Q(\xi)x(k) - x^{T}(k+j-1)Q(\xi) + x(k+j-1)Q(\xi) + x(k+j-1)\right]\right]$$

$$= E\left[(h_{2}-h_{1})x^{T}(k)Q(\xi)x(k) - \sum_{l=k+1-h_{2}}^{k-h_{1}} x^{T}(l)Q(\xi)x(l)\right].$$
(25)

Since

$$\sum_{i=k=1-h(k+1)}^{k-h_1} x^T(i) Q(\xi) x(i) - \sum_{i=k+1-h_2}^{k-h_1} x^T(i) Q(\xi) x(i) \le 0,$$
(26)

we obtain from (24) and (25) that

$$E\left[\Delta V_{2}(k) + \Delta V_{3}(k)\right]$$
  

$$\leq E\left[\left(h_{2} - h_{1} + 1\right)x^{T}(k)Q(\xi)x(k) - x^{T}(k - h(k))Q(\xi)x(k - h(k))\right].$$
(27)

Therefore, combining the inequalities (21), (27) gives

$$E\left[\Delta V\left(k\right)\right] \le E\left[\psi^{T}\left(k\right)T\left(\xi\right)\psi\left(k\right)\right],$$
(28)

where

$$\begin{split} \psi(k) &= \left[x\left(k\right)x\left(k+1\right)x\left(k-h\left(k\right)\right)\right]^{T}, \\ T\left(\xi\right) &= \begin{pmatrix} M\left(\xi\right) & S_{1}\left(\xi\right) - S_{1}\left(\xi\right)A\left(\xi\right) & -S_{1}\left(\xi\right)D\left(\xi\right) - S_{2}\left(\xi\right)A\left(\xi\right) \\ S_{1}^{T}\left(\xi\right) - A^{T}\left(\xi\right)S_{1}^{T}\left(\xi\right) & P\left(\xi\right) + S_{1}\left(\xi\right) + S_{1}^{T}\left(\xi\right) & S_{2}\left(\xi\right) - S_{1}\left(\xi\right)D\left(\xi\right) \\ -D^{T}\left(\xi\right)S_{1}^{T}\left(\xi\right) - A^{T}\left(\xi\right)S_{2}^{T}\left(\xi\right) & S_{2}^{T}\left(\xi\right) - D^{T}\left(\xi\right)S_{1}^{T}\left(\xi\right) & -Q\left(\xi\right) - S_{2}\left(\xi\right)D\left(\xi\right) - D^{T}\left(\xi\right)S_{2}^{T}\left(\xi\right) + 2\rho_{2}I \end{pmatrix}, \end{split}$$
(29)  
$$M\left(\xi\right) &= \begin{pmatrix} h_{2} - h_{1} + 1 \end{pmatrix}Q\left(\xi\right) - P\left(\xi\right) - S_{1}\left(\xi\right)A\left(\xi\right) - A\left(\xi\right)^{T}S_{1}\left(\xi\right)^{T} + 2\rho_{1}I. \end{split}$$

Let us denote that

$$M_{ij} := (h_2 - h_1 + 1) Q_i - P_i - S_{1i}A_j - A_j^T S_{1i}^T + 2\rho_1 I,$$
  

$$(S_1 A)_{ij} := S_{1j}A_i + S_{1i}A_j, \qquad (S_2 A)_{ij} := S_{2j}A_i + S_{2i}A_j,$$
  

$$(S_1 D)_{ij} := S_{1j}D_i + S_{1i}D_j, \qquad (S_2 D)_{ij} := S_{2j}D_i + S_{2i}D_j,$$

$$P_{ij} = P_i + P_j, \qquad Q_{ij} = Q_i + Q_j,$$
  

$$S_{1ij} = S_{1i} + S_{1j}, \qquad S_{2ij} = S_{2i} + S_{2j}.$$
(30)

From the convex combination of the expression of  $P(\xi)$ ,  $Q(\xi)$ ,  $S_1(\xi)$ ,  $S_2(\xi)$ ,  $A(\xi)$ ,  $D(\xi)$ , we have

$$T (\xi) = \sum_{i=1}^{p} \xi_{i}^{2} \begin{pmatrix} M_{ii} & S_{1i} - S_{1i}A_{i} & -S_{1i}D_{i} - S_{2i}A_{i} \\ S_{1i}^{T} - A_{i}^{T}S_{1i}^{T} & P_{i} + S_{1i} + S_{1i}^{T} & S_{2i} - S_{1i}D_{i} \\ -D_{i}^{T}S_{1i}^{T} - A_{i}^{T}S_{2i}^{T} & S_{2i}^{T} - D_{i}^{T}S_{1i}^{T} & -Q_{i} - S_{2i}D_{i} - D_{i}^{T}S_{2i}^{T} + 2\rho_{2}I \end{pmatrix}$$

$$+ \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \xi_{i}\xi_{j} \begin{pmatrix} M_{ij} + M_{ji} & S_{1ij} - (S_{1}A)_{ij} & -(S_{1}D)_{ij} - (S_{2}A)_{ij} \\ S_{1ij}^{T} - (A^{T}S_{1}^{T})_{ij} & P_{ij} + S_{1ij} + S_{1ij}^{T} & S_{2ij} - (S_{1}D)_{ij} \\ -(D^{T}S_{1}^{T})_{ij} - (A^{T}S_{2}^{T})_{ij} & S_{2ij}^{T} - (D^{T}S_{1}^{T})_{ij} & -Q_{ij} - (S_{2}D)_{ij} - (D^{T}S_{2}^{T})_{ij} + 2\rho_{2}I \end{pmatrix}$$

$$= \sum_{i=1}^{p} \xi_{i}^{2} \mathcal{M}_{ii} (P, Q, S_{1}, S_{2}) + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \xi_{i}\xi_{j} \left[ \mathcal{M}_{ij} (P, Q, S_{1}, S_{2}) + \mathcal{M}_{ji} (P, Q, S_{1}, S_{2}) \right].$$
(31)

Then, the conditions (i) and (ii) give

$$T(\xi) < -\sum_{i=1}^{p} \xi_{i}^{2} \mathcal{S} + \frac{2}{p-1} \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \xi_{i} \xi_{j} \mathcal{S} \le 0, \qquad (32)$$

because of Proposition 3 as

$$(p-1)\sum_{i=1}^{p}\xi_{i}^{2}-2\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\xi_{i}\xi_{j}=\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\left(\xi_{i}-\xi_{j}\right)^{2}\geq0,$$
(33)

and, hence, we finally obtain from (28) that

$$E\left[\Delta V\left(k\right)\right] \le E\left[\psi^{T}\left(k\right)T\left(\xi\right)\psi\left(k\right)\right] < 0, \quad \forall k = 0, 1, 2, \dots,$$
(34)

which together with (12) and Definition 2 implies that the system (1) is robustly stable in the mean square. This completes the proof of the theorem.  $\Box$ 

*Remark 5.* The stability conditions of Theorem 4 are more appropriate for most of real systems since it is usually impossible in practice to know exactly the delay but lower and upper bounds are always possible.

3.2. Robust Stabilization. This section deals with a stabilization problem considered in [15] for constructing a delayed feedback controller, which stabilizes the resulting closed-loop system. The robust stability condition obtained in previous section will be applied to design a time-delayed state feedback controller for the discrete-time control system described by

$$x (k + 1) = A (\xi) x (k) + B (\xi) u (k) + \sigma (x (k), x (k - h (k)), k) \omega (k),$$
(35)  
$$k = 0, 1, 2, ...,$$

where  $u(k) \in \mathbb{R}^n$  is the control input, and the system matrices are subjected to uncertainties and belong to the polytope  $\Omega$ given by

$$\Omega = \left\{ [A, B] (\xi) := \sum_{i=1}^{p} \xi_i [A_i, B_i], \sum_{i=1}^{p} \xi_i = 1, \xi_i \ge 0 \right\},$$
(36)

where  $A_i, B_i, i = 1, 2, ..., p$ , are given constant matrices with appropriate dimensions. As in [8], we consider a parameter-dependent delayed feedback control law

$$u(k) = F(\xi) x(k - h(k)), \quad k = -h_2, \dots, 0, \quad (37)$$

where h(k) is the time-varying delay function satisfying  $0 < h_1 \le h(k) \le h_2$ , and  $F(\xi)$  is the controller gain to be determined. Applying the feedback controller (37) to the system (35), the closed-loop time-delay system is

$$x (k + 1) = A (\xi) x (k) + B (\xi) F (\xi) x (k - h (k)) + \sigma (x (k), x (k - h (k)), k) \omega (k),$$
(38)  
$$k = 0, 1, 2, ...$$

*Definition 6.* The system (35) is robustly stabilizable in the mean square if there is a delayed feedback control (37) such that the closed-loop delay system (38) is robustly stable in the mean square.

Let

$$\mathcal{M}_{ij}(P,Q,S_{1}) = \begin{pmatrix} (h_{2} - h_{1} + 1)Q_{i} - P_{i} - S_{1i}A_{j} - A_{j}^{T}S_{1i}^{T} + 2\rho_{1}I & S_{1i} - S_{1i}A_{j} & -P_{i} - S_{1i}A_{j} \\ S_{1i}^{T} - A_{j}^{T}S_{1i}^{T} & P_{i} + S_{1i} + S_{1i}^{T} & S_{1i} - P_{i} \\ -P_{i} - A_{j}^{T}S_{1i}^{T} & S_{1i}^{T} - P_{i} & -Q_{i} - P_{i} - P_{i} + 2\rho_{2}I \end{pmatrix},$$

$$\mathcal{S} = \begin{pmatrix} S & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The following theorem can be derived from Theorem 4.

**Theorem 7.** *The system* (35) *is robustly stabilizable in the mean square by the delayed feedback control* (37), *where* 

$$F(\xi) = B^{T}(\xi) \left[ B(\xi) B^{T}(\xi) \right]^{-1} S_{1}^{T}(\xi) \left[ S_{1}(\xi) S_{1}^{T}(\xi) \right]^{-1} P(\xi),$$
(40)

if there exist symmetric matrices  $P_i > 0$ ,  $Q_i > 0$ , i = 1, 2, ..., p, and constant matrices  $S_{1i}$ , i = 1, 2, ..., p,  $S \ge 0$ , satisfying the following LMIs:

(i) 
$$\mathcal{M}_{ii}(P,Q,S_1) + \mathcal{S} < 0, i = 1, 2, ..., p;$$
  
(ii)  $\mathcal{M}_{ij}(P,Q,S_1) + \mathcal{M}_{ji}(P,Q,S_1) - (2/(p-1))\mathcal{S} < 0, i = 1, 2, ..., p - 1; j = i + 1, ..., p.$ 

*Proof.* Taking  $S_{1i} = S_{2i}$  and using the feedback control (37), the closed-loop system becomes system  $(\Sigma_{\xi})$ , where  $D(\xi) = B(\xi)F(\xi) = S_1^T(\xi)[S_1(\xi)S_1^T(\xi)]^{-1}P(\xi)$ . Since  $S_1(\xi)D(\xi) = P(\xi)$ , the robust stability condition of the closed-loop system (38), by Theorem 4, is immediately derived.

*Remark* 8. The stabilization conditions of Theorem 7 are more appropriate for most of real systems since it is usually impossible in practice to know exactly the delay but lower and upper bounds are always possible.

#### 4. Numerical Examples

To illustrate the effectiveness of the previous theoretical results, we consider the following numerical examples.

*Example 9* (robust stability). Consider system  $\Sigma_{\xi}$  for p = 2, where the delay function h(k) is given by

$$h(k) = 1 + 28 \sin^2 \frac{k\pi}{2}, \quad k = 0, 1, 2, \dots,$$
  

$$A_1 = \begin{pmatrix} -30.5 & 1\\ 2 & -3.5 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -35.5 & 1\\ 3 & -4.5 \end{pmatrix}, \quad (41)$$
  

$$D_1 = \begin{pmatrix} -1.5 & 0.1\\ 0.4 & -2.15 \end{pmatrix}, \quad D_2 = \begin{pmatrix} -2.5 & 0.2\\ 0.3 & -1.85 \end{pmatrix}.$$

By using the LMI Toolbox in MATLAB, the LMIs (i) and (ii) of Theorem 4 are feasible with  $h_1 = 1$ ,  $h_2 = 29$ ,  $\rho_1 = 0.011$ ,  $\rho_2 = 0.015$ , and we use the condition in the Theorem 4 for this example. The solutions of LMI verify as follow of the form

$$P_{1} = \begin{pmatrix} 4.6120 & 0.2565 \\ 0.2565 & 3.3703 \end{pmatrix}, \qquad P_{2} = \begin{pmatrix} 2.9556 & 0.0381 \\ 0.0381 & 3.5256 \end{pmatrix},$$

$$Q_{1} = \begin{pmatrix} 0.1402 & 0.0109 \\ 0.0109 & 0.0145 \end{pmatrix}, \qquad Q_{2} = \begin{pmatrix} 0.3550 & 0.0101 \\ 0.0101 & 0.2101 \end{pmatrix},$$

$$S_{11} = \begin{pmatrix} -0.0596 & -0.0430 \\ 0.0031 & 0.0453 \end{pmatrix},$$

$$S_{12} = \begin{pmatrix} -0.0197 & -0.0095 \\ 0.0045 & 0.0375 \end{pmatrix},$$

$$S_{21} = \begin{pmatrix} 0.0006 & 0.0250 \\ -0.0029 & -0.1615 \end{pmatrix},$$

$$S_{22} = \begin{pmatrix} -0.0002 & 0.0133 \\ -0.0030 & -0.1228 \end{pmatrix},$$

$$S = \begin{pmatrix} 2.0759 & 0.0459 \\ 0.0459 & 1.3271 \end{pmatrix}.$$
(42)

Therefore, the system is robustly stable.

*Example 10* (robust stabilization). Consider system (35) for p = 2, where the delay function h(k) is given by

$$h(k) = 1 + 34 \sin^2 \frac{k\pi}{2}, \quad k = 0, 1, 2, \dots,$$

$$A_1 = \begin{pmatrix} -30.5 & 1\\ 2 & -3.5 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -35.5 & 1\\ 3 & -4.5 \end{pmatrix}, \quad (43)$$

$$B_1 = \begin{pmatrix} -1.5 & 0.1\\ 0.4 & -2.15 \end{pmatrix}, \quad B_2 = \begin{pmatrix} -2.5 & 0.2\\ 0.3 & -1.85 \end{pmatrix}.$$

(39)

By using the LMI Toolbox in MATLAB, the LMIs (i) and (ii) of Theorem 7 are feasible with  $h_1 = 1$ ,  $h_2 = 35$ ,  $\rho_1 = 0.011$ ,  $\rho_2 = 0.015$ , and we use the condition in the Theorem 7 for this example. The solutions of LMI verify as follow of the form

$$P_{1} = \begin{pmatrix} 1.3886 & -0.0760 \\ -0.0760 & 1.3559 \end{pmatrix}, \qquad P_{2} = \begin{pmatrix} 1.6286 & 0.0649 \\ 0.0649 & 1.5243 \end{pmatrix},$$
$$Q_{1} = \begin{pmatrix} 0.0097 & -0.0048 \\ -0.0048 & 0.0057 \end{pmatrix},$$
$$Q_{2} = \begin{pmatrix} 0.0728 & -0.0159 \\ -0.0159 & 0.0621 \end{pmatrix},$$

$$S_{11} = \begin{pmatrix} -0.0274 & 0.0827 \\ -0.0133 & -0.2222 \end{pmatrix},$$
  

$$S_{12} = \begin{pmatrix} -0.0209 & 0.0619 \\ -0.0226 & -0.1942 \end{pmatrix},$$
  

$$S = \begin{pmatrix} 0.5954 & -0.0672 \\ -0.0672 & 0.5469 \end{pmatrix}.$$
(44)

Therefore, the system is robustly stabilizable with the feedback control

$$\begin{split} u(k) &= B^{T}(\xi) \left[ B(\xi) B^{T}(\xi) \right]^{-1} S_{1}^{T}(\xi) \left[ S_{1}(\xi) S_{1}^{T}(\xi) \right]^{-1} P(\xi) x(k-h(k)) \\ &= \left( \xi_{1}B_{1} + \xi_{2}B_{2} \right)^{T} \left[ \left( \xi_{1}B_{1} + \xi_{2}B_{2} \right) \left( \xi_{1}B_{1} + \xi_{2}B_{2} \right)^{T} \right]^{-1} \\ &\times \left( \xi_{1}S_{11} + \xi_{2}S_{12} \right)^{T} \left[ \left( \xi_{1}S_{11} + \xi_{2}S_{12} \right) \left( \xi_{1}S_{11} + \xi_{2}S_{12} \right)^{T} \right]^{-1} \left( \xi_{1}P_{1} + \xi_{2}P_{2} \right) (\xi) x(k-h(k)) \right] \\ &= \begin{pmatrix} -1.5\xi_{1} - 2.5\xi_{2} & 0.4\xi_{1} + 0.3\xi_{2} \\ 0.1\xi_{1} + 0.2\xi_{2} & -2.15\xi_{1} - 1.85\xi_{2} \end{pmatrix} \times \begin{pmatrix} -1.5\xi_{1} - 2.5\xi_{2} & 0.4\xi_{1} + 0.3\xi_{2} \\ 0.1\xi_{1} + 0.2\xi_{2} & -2.15\xi_{1} - 1.85\xi_{2} \end{pmatrix}^{-1} \\ &\times \begin{pmatrix} -1.5\xi_{1} - 2.5\xi_{2} & 0.1\xi_{1} + 0.2\xi_{2} \\ 0.4\xi_{1} + 0.3\xi_{2} & -2.15\xi_{1} - 1.85\xi_{2} \end{pmatrix}^{-1} \times \begin{pmatrix} -0.0274\xi_{1} - 0.0209\xi_{2} & -0.0133\xi_{1} - 0.0226\xi_{2} \\ 0.0827\xi_{1} + 0.0619\xi_{2} & -0.2222\xi_{1} - 0.1942\xi_{2} \end{pmatrix}^{-1} \\ &\times \begin{pmatrix} -0.0274\xi_{1} - 0.0209\xi_{2} & -0.0133\xi_{1} - 0.0226\xi_{2} \\ 0.0827\xi_{1} + 0.0619\xi_{2} & -0.222\xi_{1} - 0.1942\xi_{2} \end{pmatrix}^{-1} \times \begin{pmatrix} -0.0274\xi_{1} - 0.0209\xi_{2} & 0.0827\xi_{1} + 0.0619\xi_{2} \\ -0.0133\xi_{1} - 0.0226\xi_{2} & -0.2222\xi_{1} - 0.1942\xi_{2} \end{pmatrix}^{-1} \\ &\times \begin{pmatrix} 1.3886\xi_{1} + 1.6286\xi_{2} & -0.0760\xi_{1} + 0.0649\xi_{2} \\ -0.0760\xi_{1} + 0.0649\xi_{2} & 1.3559\xi_{1} + 1.5243\xi_{2} \end{pmatrix}^{-1} \\ &\times \begin{pmatrix} -2.0829\xi_{1}^{2} - 5.9144\xi_{1}\xi_{2} - 4.0715\xi_{2}^{2} & -0.0304\xi_{1}^{2} + 0.0260\xi_{1}\xi_{2} + 0.0195\xi_{2}^{2} \\ -0.0076\xi_{1}^{2} - 0.0087\xi_{1}\xi_{2} + 0.0128\xi_{2}^{2} & -2.9152\xi_{1}^{2} - 5.7856\xi_{1}\xi_{2} - 2.8200\xi_{2}^{2} \end{pmatrix} x(k-h(k)). \end{split}$$

Therefore, the feedback delayed controller is

$$u_{1}(k) = \begin{bmatrix} -2.0829\xi_{1}^{2} - 5.9144\xi_{1}\xi_{2} - 4.0715\xi_{2}^{2} \end{bmatrix} \\ \times x_{1}(k - h(k)) \\ + \begin{bmatrix} -0.0304\xi_{1}^{2} + 0.0260\xi_{1}\xi_{2} + 0.0195\xi_{2}^{2} \end{bmatrix} \\ \times x_{2}(k - h(k)),$$

$$u_{2}(k) = \begin{bmatrix} -0.0076\xi_{1}^{2} - 0.0087\xi_{1}\xi_{2} + 0.0128\xi_{2}^{2} \end{bmatrix} \\ \times x_{1}(k - h(k)) \\ + \begin{bmatrix} -2.9152\xi_{1}^{2} - 5.7856\xi_{1}\xi_{2} - 2.8200\xi_{2}^{2} \end{bmatrix} \\ \times x_{2}(k - h(k)).$$

$$(46)$$

## 5. Conclusion

In this paper, new delay-dependent mean square robust stability conditions for linear polytopic delay-difference stochastic equations with interval time-varying delays have been presented in terms of LMIs. An application to mean square robust stabilization of discrete stochastic control systems with time-delayed feedback controllers has been studied. Numerical examples have been given to demonstrate the effectiveness of the proposed conditions.

#### Acknowledgments

This work was supported by the Office of Agricultural Research and Extension Maejo University, the Thailand Research Fund Grant, the Higher Education Commission, and Faculty of Science, Maejo University, Thailand. The first author is supported by the Center of Excellence in Mathematics, Thailand, and Commission for Higher Education, Thailand. The authors thank anonymous reviewers for valuable comments and suggestions, which allowed them to improve the paper.

## References

- [1] R. P. Agarwal, *Difference Equations and Inequalities*, vol. 228, Marcel Dekker Inc., New York, NY, USA, 2nd edition, 2000.
- [2] W. H. Chen, Z. H. Guan, and X. Lu, "Delay-dependent guaranteed cost control for uncertain discrete-time systems with

delay," *IEE Proceedings: Control Theory and Applications*, vol. 150, no. 4, pp. 412–416, 2003.

- [3] S. Elaydi and I. Győri, "Asymptotic theory for delay difference equations," *Journal of Difference Equations and Applications*, vol. 1, no. 2, pp. 99–116, 1995.
- [4] V. Kolmanovskii and A. Myshkis, Applied Theory of Functional Differential Equations, Springer, Berlin, Germany, 1992.
- [5] O. M. Kwon and J. H. Park, "Exponential stability of uncertain dynamic systems including state delay," *Applied Mathematics Letters*, vol. 19, no. 9, pp. 901–907, 2006.
- [6] W. J. Mao and J. Chu, "D-stability and D-stabilization of linear discrete time-delay systems with polytopic uncertainties," *Automatica*, vol. 45, no. 3, pp. 842–846, 2009.
- [7] P. T. Nam, H. M. Hien, and V. N. Phat, "Asymptotic stability of linear state-delayed neutral systems with polytope type uncertainties," *Dynamic Systems and Applications*, vol. 19, no. 1, pp. 63–72, 2010.
- [8] V. N. Phat and J. Y. Park, "On the Gronwall inequality and asymptotic stability of nonlinear discrete systems with multiple delays," *Dynamic Systems and Applications*, vol. 10, no. 4, pp. 577–588, 2001.
- [9] T. L. Hsien and C. H. Lee, "Exponential stability of discrete time uncertain systems with time-varying delay," *Journal of the Franklin Institute*, vol. 332, no. 4, pp. 479–489, 1995.
- [10] D. H. Ji, J. H. Park, W. J. Yoo, and S. C. Won, "Robust memory state feedback model predictive control for discretetime uncertain state delayed systems," *Applied Mathematics and Computation*, vol. 215, no. 6, pp. 2035–2044, 2009.
- [11] G. Rajchakit, T. Rojsiraphisal, and M. Rajchakit, "Robust stability and stabilization of uncertain switched discrete-time systems," *Advances in Difference Equations*, vol. 2012, article 134, 2012.
- [12] E. K. Boukas, "State feedback stabilization of nonlinear discretetime systems with time-varying time delay," *Nonlinear Analysis. Theory, Methods & Applications.*, vol. 66, no. 6, pp. 1341–1350, 2007.
- [13] H. Gao and T. Chen, "New results on stability of discrete-time systems with time-varying state delay," *IEEE Transactions on Automatic Control*, vol. 52, no. 2, pp. 328–334, 2007.
- [14] X. Jiang, Q. L. Han, and X. Yu, "Stability criteria for linear discrete-time systems with interval-like time-varying delay," in *Proceedings of the American Control Conference (ACC '05)*, pp. 2817–2822, June 2005.
- [15] B. Zhang, S. Xu, and Y. Zou, "Improved stability criterion and its applications in delayed controller design for discrete-time systems," *Automatica*, vol. 44, no. 11, pp. 2963–2967, 2008.
- [16] M. Yu, L. Wang, and T. Chu, "Robust stabilization of discretetime systems with time-varying delays," in *Proceedings of the American Control Conference (ACC '05)*, pp. 3435–3440, Portland, Ore, USA, June 2005.
- [17] K. Ratchagit and V. N. Phat, "Stability criterion for discretetime systems," *Journal of Inequalities and Applications*, vol. 2010, Article ID 201459, 6 pages, 2010.
- [18] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel, "Qualitative analysis of discrete-time switched systems," in *Proceedings of the American Control Conference (ACC '02)*, pp. 1880–1885, May 2002.
- [19] M. Rajchakit, P. Niamsup, and G. Rajchakit, "A switching rule for exponential stability of switched recurrent neural networks with interval time-varying delay," *Advances in Difference Equations*, vol. 2013, article 44, 2013.

- [20] K. Ratchagit, "A switching rule for the asymptotic stability of discrete-time systems with convex polytopic uncertainties," *Asian-European Journal of Mathematics*, vol. 5, no. 2, Article ID 1250025, 12 pages, 2012.
- [21] M. Rajchakit and G. Rajchakit, "Mean square exponential stability of stochastic switched system with interval timevarying delays," *Abstract and Applied Analysis*, vol. 2012, Article ID 623014, 12 pages, 2012.
- [22] W. A. Zhang and L. Yu, "Stability analysis for discrete-time switched time-delay systems," *Automatica*, vol. 45, no. 10, pp. 2265–2271, 2009.
- [23] V. N. Phat and K. Ratchagit, "Stability and stabilization of switched linear discrete-time systems with interval time-varying delay," *Nonlinear Analysis: Hybrid Systems*, vol. 5, no. 4, pp. 605–612, 2011.
- [24] M. O. Abou Al-Ola, K. Fujimoto, and T. Yoshinaga, "Common Lyapunov function based on Kullback-Leibler divergence for a switched nonlinear system," *Mathematical Problems in Engineering*, vol. 2011, Article ID 723509, 12 pages, 2011.
- [25] Y. He, M. Wu, J. H. She, and G. P. Liu, "Parameter-dependent Lyapunov functional for stability of time-delay systems with polytopic-type uncertainties," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 828–832, 2004.
- [26] D. Henrion, D. Arzelier, D. Peaucelle, and M. Šebek, "LMI condition for robust stability of polynomial matrix polytopes," *Automatica*, vol. 37, no. 3, pp. 461–468, 2001.
- [27] D. F. Coutinho, M. Fu, and A. Trofino, "Robust analysis and control for a class of uncertain nonlinear discrete-time systems," *Systems and Control Letters*, vol. 53, no. 5, pp. 377–393, 2004.
- [28] S. W. Kau, Y. Liu, L. Hang, C. H. Lee, C. H. Fang, and L. Lee, "A new LMI condition for robust stability of discrete-time uncertain systems," *Applied Mathematics and Computation*, vol. 215, pp. 2035–2044, 2009.
- [29] M. Rajchakit and G. Rajchakit, "Mean square robust stability of stochastic switched discrete-time systems with convex polytopic uncertainties," *Journal of Inequalities and Applications*, vol. 2012, article 135, 2012.
- [30] B. Du and X. Zhang, "Delay-dependent stability analysis and synthesis for uncertain impulsive switched system with mixed delays," *Discrete Dynamics in Nature and Society*, vol. 2011, Article ID 381571, 9 pages, 2011.
- [31] M. De la Sen and A. Ibeas, "Stability results of a class of hybrid systems under switched continuous-time and discretetime control," *Discrete Dynamics in Nature and Society*, vol. 2009, Article ID 315713, 28 pages, 2009.
- [32] M. de la Sen and A. Ibeas, "Stability results for switched linear systems with constant discrete delays," *Mathematical Problems in Engineering*, vol. 2008, Article ID 543145, 28 pages, 2008.
- [33] M. Rajchakit and G. Rajchakit, "LMI approach to robust stability and stabilization of nonlinear uncertain discrete-time systems with convex polytopic uncertainties," *Advances in Difference Equations*, vol. 2012, 2012, article 106.
- [34] J. Li, W. Wu, J. Yuan, Q. Tan, and X. Yin, "Delay-dependent stability criterion of arbitrary switched linear systems with time-varying delay," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 347129, 16 pages, 2010.
- [35] L. Zhang, H. Li, and Y. Chen, "Robust stability analysis and synthesis for switched discrete-time systems with time delay," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 408105, 19 pages, 2010.

[36] M. De La Sen, "On the characterization of hankel and toeplitz operators describing switched linear dynamic systems with point delays," *Abstract and Applied Analysis*, vol. 2009, Article ID 670314, 34 pages, 2009.