

Complexity

Computational Intelligence in Modeling Complex Systems and Solving Complex Problems

Lead Guest Editor: László T. Kóczy

Guest Editors: Jesus Medina, Marek Reformat, Jin H. Yoon, and Kevin Wong





Computational Intelligence in Modeling Complex Systems and Solving Complex Problems

Complexity

Computational Intelligence in Modeling Complex Systems and Solving Complex Problems

Lead Guest Editor: László T. Kóczy

Guest Editors: Jesus Medina, Marek Reformat, Jin H. Yoon,
and Kok Wai Wong



Copyright © 2019 Hindawi. All rights reserved.

This is a special issue published in “Complexity.” 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

- José A. Acosta, Spain
Carlos F. Aguilar-Ibáñez, Mexico
Mojtaba Ahmadiéh Khanesar, UK
Tarek Ahmed-Ali, France
Alex Alexandridis, Greece
Basil M. Al-Hadithi, Spain
Juan A. Almendral, Spain
Diego R. Amancio, Brazil
David Arroyo, Spain
Mohamed Boutayeb, France
Átila Bueno, Brazil
Arturo Buscarino, Italy
Guido Caldarelli, Italy
Eric Campos-Canton, Mexico
Mohammed Chadli, France
Émile J. L. Chappin, Netherlands
Diyi Chen, China
Yu-Wang Chen, UK
Giulio Cimini, Italy
Danilo Comminiello, Italy
Sara Dadras, USA
Sergey Dashkovskiy, Germany
Manlio De Domenico, Italy
Pietro De Lellis, Italy
Albert Diaz-Guilera, Spain
Thach Ngoc Dinh, France
Jordi Duch, Spain
Marcio Eisencraft, Brazil
Joshua Epstein, USA
Mondher Farza, France
Thierry Floquet, France
Mattia Frasca, Italy
José Manuel Galán, Spain
Lucia Valentina Gambuzza, Italy
Bernhard C. Geiger, Austria
Carlos Gershenson, Mexico
Peter Giesl, UK
Sergio Gómez, Spain
Lingzhong Guo, UK
Xianggui Guo, China
Sigurdur F. Hafstein, Iceland
Chittaranjan Hens, Israel
Giacomo Innocenti, Italy
Sarangapani Jagannathan, USA
Mahdi Jalili, Australia
Jeffrey H. Johnson, UK
M. Hassan Khooban, Denmark
Abbas Khosravi, Australia
Toshikazu Kuniya, Japan
Vincent Labatut, France
Lucas Lacasa, UK
Guang Li, UK
Qingdu Li, Germany
Chongyang Liu, China
Xiaoping Liu, Canada
Xinzhi Liu, Canada
Rosa M. Lopez Gutierrez, Mexico
Vittorio Loreto, Italy
Noureddine Manamanni, France
Didier Maquin, France
Eulalia Martínez, Spain
Marcelo Messias, Brazil
Ana Meštrović, Croatia
Ludovico Minati, Japan
Ch. P. Monterola, Philippines
Marcin Mrugalski, Poland
Roberto Natella, Italy
Sing Kiong Nguang, New Zealand
Nam-Phong Nguyen, USA
B. M. Ombuki-Berman, Canada
Irene Otero-Muras, Spain
Yongping Pan, Singapore
Daniela Paolotti, Italy
Cornelio Posadas-Castillo, Mexico
Mahardhika Pratama, Singapore
Luis M. Rocha, USA
Miguel Romance, Spain
Avimanyu Sahoo, USA
Matilde Santos, Spain
Josep Sardanyés Cayuela, Spain
Ramaswamy Savitha, Singapore
Hiroki Sayama, USA
Michele Scarpiniti, Italy
Enzo Pasquale Scilingo, Italy
Dan Selișteanu, Romania
Dehua Shen, China
Dimitrios Stamovlasis, Greece
Samuel Stanton, USA
Roberto Tonelli, Italy
Shahadat Uddin, Australia
Gaetano Valenza, Italy
Dimitri Volchenkov, USA
Christos Volos, Greece
Zidong Wang, UK
Yan-Ling Wei, Singapore
Honglei Xu, Australia
Yong Xu, China
Xinggang Yan, UK
Baris Yuçe, UK
Massimiliano Zanin, Spain
Hassan Zargarzadeh, USA
Rongqing Zhang, USA
Xianming Zhang, Australia
Xiaopeng Zhao, USA
Quanmin Zhu, UK

Contents

Computational Intelligence in Modeling Complex Systems and Solving Complex Problems

Laszlo T. Koczy , Jesus Medina , Marek Reformat, Kok Wai Wong, and Jin Hee Yoon
Editorial (6 pages), Article ID 7606715, Volume 2019 (2019)

Simulations of Higher-Order Protein Organizations Using a Fuzzy Framework

B. Tüü-Szabó , L. T. Kóczy , and M. Fuxreiter 
Research Article (10 pages), Article ID 6360846, Volume 2018 (2019)

An Improved Differential Evolution Algorithm for a Multicommodity Location-Inventory Problem with False Failure Returns

Congdong Li , Hao Guo , Ying Zhang , Shuai Deng , and Yu Wang
Research Article (13 pages), Article ID 1967398, Volume 2018 (2019)

Simulation-Based Optimization on the System-of-Systems Model via Model Transformation and Genetic Algorithm: A Case Study of Network-Centric Warfare

Bong Gu Kang , Seon Han Choi , Se Jung Kwon, Jun Hee Lee , and Tag Gon Kim 
Research Article (15 pages), Article ID 4521672, Volume 2018 (2019)

A Binary Cuckoo Search Big Data Algorithm Applied to Large-Scale Crew Scheduling Problems

José García , Francisco Altimiras , Alvaro Peña, Gino Astorga, and Oscar Peredo
Research Article (15 pages), Article ID 8395193, Volume 2018 (2019)

A Nonlinear Integer Programming Model for Integrated Location, Inventory, and Routing Decisions in a Closed-Loop Supply Chain

Hao Guo , Congdong Li , Ying Zhang , Chunnan Zhang, and Yu Wang
Research Article (17 pages), Article ID 2726070, Volume 2018 (2019)

Regression and ANN Models for Electronic Circuit Design

M. I. Dieste-Velasco , M. Diez-Mediavilla , and C. Alonso-Tristán 
Research Article (9 pages), Article ID 7379512, Volume 2018 (2019)

Integrated Feature Selection of ARIMA with Computational Intelligence Approaches for Food Crop Price Prediction

Yuehjen E. Shao  and Jun-Ting Dai
Research Article (17 pages), Article ID 1910520, Volume 2018 (2019)

About Extracting Dynamic Information of Unknown Complex Systems by Neural Networks

Eloy Irigoyen , Antonio Javier Barragán, Mikel Larrea , and José Manuel Andújar 
Research Article (12 pages), Article ID 3671428, Volume 2018 (2019)

Some Hesitant Fuzzy Linguistic Muirhead Means with Their Application to Multiattribute Group Decision-Making

Jun Wang , Runtong Zhang , Xiaomin Zhu , Yuping Xing, and Borut Buchmeister
Research Article (16 pages), Article ID 5087851, Volume 2018 (2019)

Communication Analysis of Network-Centric Warfare via Transformation of System of Systems Model into Integrated System Model Using Neural Network

Bong Gu Kang , Kyung-Min Seo , and Tag Gon Kim 

Research Article (16 pages), Article ID 6201356, Volume 2018 (2019)

Systematic Approach to Optimization for Protection Against Intentional Ultrashort Pulses Based on Multiconductor Modal Filters

Anton O. Belousov  and Talgat R. Gazizov 

Research Article (15 pages), Article ID 5676504, Volume 2018 (2019)

An Improved MOEA/D Based on Reference Distance for Software Project Portfolio Optimization

Jing Xiao , Jing-Jing Li , Xi-Xi Hong, Min-Mei Huang , Xiao-Min Hu , Yong Tang, and Chang-Qin Huang 

Research Article (16 pages), Article ID 3051854, Volume 2018 (2019)

Improved Method for Predicting the Performance of the Physical Links in Telecommunications Access Networks

Ferenc Lilik , Szilvia Nagy , and László T. Kóczy 

Research Article (14 pages), Article ID 3685927, Volume 2018 (2019)

Trust-Based Situation Awareness: Comparative Analysis of Agent-Based and Population-Based Modeling

Zara Nasar  and Syed Waqar Jaffry 

Research Article (17 pages), Article ID 9540726, Volume 2018 (2019)

Improved Optimization for Wastewater Treatment and Reuse System Using Computational Intelligence

Zong Woo Geem , Sung Yong Chung, and Jin-Hong Kim 

Research Article (8 pages), Article ID 2480365, Volume 2018 (2019)

The Intuitionistic Fuzzy Linguistic Cosine Similarity Measure and Its Application in Pattern Recognition

Donghai Liu , Xiaohong Chen, and Dan Peng

Research Article (11 pages), Article ID 9073597, Volume 2018 (2019)

A Hybrid Approach for Modular Neural Network Design Using Intercriteria Analysis and Intuitionistic Fuzzy Logic

Sotir Sotirov , Evdokia Sotirova, Vassia Atanassova, Krassimir Atanassov, Oscar Castillo , Patricia Melin, Todor Petkov, and Stanimir Surchev

Research Article (11 pages), Article ID 3927951, Volume 2018 (2019)

Legendre Cooperative PSO Strategies for Trajectory Optimization

Lei Liu , Yongji Wang , Fuqiang Xie, and Jiashi Gao

Research Article (13 pages), Article ID 5036791, Volume 2018 (2019)

Extension of the Multi-TP Model Transformation to Functions with Different Numbers of Variables

Péter Baranyi 

Research Article (9 pages), Article ID 8546976, Volume 2018 (2019)

The Spiral Discovery Network as an Automated General-Purpose Optimization Tool

Adam B. Csapo 

Research Article (8 pages), Article ID 1947250, Volume 2018 (2019)

Effective Evolutionary Multilabel Feature Selection under a Budget Constraint

Jaesung Lee , Wangduk Seo , and Dae-Won Kim 

Research Article (14 pages), Article ID 3241489, Volume 2018 (2019)

Experimental Verification of Optimized Multiscroll Chaotic Oscillators Based on Irregular Saturated Functions

J. M. Muñoz-Pacheco , D. K. Guevara-Flores, O. G. Félix-Beltrán, E. Tlelo-Cuautle ,
J. E. Barradas-Guevara, and C. K. Volos

Research Article (17 pages), Article ID 3151840, Volume 2018 (2019)

Improved Hybrid Fireworks Algorithm-Based Parameter Optimization in High-Order Sliding Mode Control of Hypersonic Vehicles

Xiaomeng Yin, Xing Wei, Lei Liu , and Yongji Wang

Research Article (16 pages), Article ID 9098151, Volume 2018 (2019)

An Improved Test Selection Optimization Model Based on Fault Ambiguity Group Isolation and Chaotic Discrete PSO

Xiaofeng Lv, Deyun Zhou, Yongchuan Tang , and Ling Ma

Research Article (10 pages), Article ID 3942723, Volume 2018 (2019)

Computational Analysis of Complex Population Dynamical Model with Arbitrary Order

Fazal Haq , Kamal Shah , Ghaus ur Rahman, Yongjin Li , and Muhammad Shahzad 

Research Article (8 pages), Article ID 8918541, Volume 2018 (2019)

Parameter Tuning for Local-Search-Based Metaheuristic Methods

Guillermo Cabrera-Guerrero, Carolina Lagos, Carolina Castañeda, Franklin Johnson, Fernando Paredes,
and Enrique Cabrera

Research Article (15 pages), Article ID 1702506, Volume 2017 (2019)

Editorial

Computational Intelligence in Modeling Complex Systems and Solving Complex Problems

Laszlo T. Koczy ¹, **Jesus Medina** ², **Marek Reformat**,³
Kok Wai Wong,⁴ and **Jin Hee Yoon**⁵

¹*Budapest University of Technology and Economics, Budapest, Hungary*

²*University of Cadiz, Cadiz, Spain*

³*University of Alberta, Edmonton, AB, Canada*

⁴*Murdoch University, Perth, WA, Australia*

⁵*Sejong University, Seoul, Republic of Korea*

Correspondence should be addressed to Laszlo T. Koczy; koczy@tmit.bme.hu

Received 2 December 2018; Accepted 2 December 2018; Published 2 January 2019

Copyright © 2019 Laszlo T. Koczy 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.

In the special issue on computational intelligence (CI) and complex systems a plethora of approaches is presented on how to apply computation intelligence in modeling, control, decision support, and optimization of complex problems. The three main components of CI are fuzzy systems, evolutionary and population based algorithms, and artificial neural networks. In addition, there are other related techniques, among others, chaos theory and subjective probability. Often, the combinations and hybrids of these methods, sometimes complemented by classic mathematical or modeling tools, turn out to be most efficient in the solution of real life problems. In the series of 26 papers, illustrations for almost all of these methods may be found.

What is the main goal and target in these research problems? Several years ago the Lead Guest Editor published a study on how to use fuzzy rule based systems as tools in solving what was called (maybe, in a somewhat exaggerating way) the “Key Problem of Engineering,” even though “Key Problem of Engineering” is not a term accepted by consensus in the relevant literature. Moreover, the term Engineering has many different interpretations, the narrowest one referring to technological sciences and disciplines only, while in various broader interpretations it includes computer science, agricultural engineering, social engineering, and certain aspects of economic models. In the present special issue, Engineering may be replaced by the concepts “Applied Problems,” “Real Life Applications,” or similar.

The essential point is that problems arising in various real life contexts have a number of common features, namely, their high complexity and the requirement of solving the problem with “reasonable quality” from the application side. The behavior of this kind of models was addressed in a form reduced to optimizing fuzzy rule based models from the point of view of practical applicability, which was called “Fuzzy Cat and Mouse Problem”: a theoretical experiment where the “fuzzy cat” was the implementation of a simple CI approach (the model and algorithm representing a Mamdani-type fuzzy system). In this example, the problem to be solved is to catch a mouse following a very simple and mechanical algorithm:

- (1) The cat takes a “picture” of the position and dynamic features of the mouse.
- (2) The rule base in the “head of the cat” is calculating the likely future position of the mouse at the time of the cat grasping it. (Because of the uncertainty involved, this future position may be given only by an estimated area.)
- (3) The cat searches the estimated future position area of the mouse systematically until it catches the mouse.

Essentially, two time periods will add together when the sum time of catching the mouse is calculated:

- (i) the time of running the Mamdani-system for the determination of the area containing the future position of the mouse,
- (ii) the time searching that area (exhaustively).

This sum time should be minimized. The dilemma is the following: if the model is very refined and precise, its evaluation, i.e., the calculation of the expected future position of the mouse, takes a longer time, and thus, the uncertainty of the mouse's position becomes bigger, and so the area to be searched becomes larger. However, within the limits of this uncertainty, the estimated future position (the "area center") will be calculated more precisely, and thus, the resulting uncertainty of the future position area, which is identical to the area to be searched in step 3 of the algorithm, will be determined by the delicate balance between the fineness and precision of the prediction model and the speed of calculating the result, as obviously, the smaller the area to be searched, the faster the prediction model produces a result.

All other complex problems are similar in the sense that the balance of resource intensity and solution quality must satisfy a preliminarily fixed goal function (where parameters may set the relative importance of the two goal components). The entirety of CI methods, along with hybrid approaches, where CI is combined with traditional mathematical and statistical methods, and computer science techniques, form a nice toolkit from where such solutions satisfy the user or the presenter of a problem to a high degree.

It is not easy to tell anything new about the concept of complexity in the Complexity Journal. A long series of articles having been published in the past present a colorful and rich pool of complex problems with various approaches to the definition of complexity. They agree that the high need of resources (especially time and space) makes them intractable in the mathematical sense (usually they fall in the NP-hard class), but they often involve nondeterministic behavior and all types of vagueness and uncertainty. The reader will judge if all the articles in the special issue conform to this definition.

Next, a very brief description of all the papers will follow. We arranged the papers according to the main CI approach applied, wherever it was possible, while the hybrid approaches were listed between the two respective main approaches combined. Finally, the few "outliers," the articles on using less frequently applied CI techniques, were listed at the end of the issue.

In this collection of studies, the most frequently appearing methodological approach is unquestionably the wide and still continuously increasing group of evolutionary and population based algorithms, which are applied for optimization and search. To use a pun, there is a permanent evolution of the evolutionary algorithms going on, and in some applications even the more classic approaches themselves produce sometimes surprisingly good results. It is obvious that highly complex (NP-hard, exponential complexity) problems cannot be solved generally by any exact mathematical method. This is where the random element inherently occurring in the evolutionary approaches gets its importance: both in the mutation type and in the gene/chromosome transfer operations there is

a random component which allows global movements in the optimization space. Although in the case of the evolutionary optimization algorithms, no guarantee of an exact or, even, almost exact optimization may be given, often a rather good solution is achieved by these meta-heuristic techniques, and simulation based evidence supports the expectations of these being always well performing techniques. Next the papers classified into this category will be listed.

X. Yin et al. propose in the paper **Improved Hybrid Fireworks Algorithm-Based Parameter Optimization in High-Order Sliding Mode Control of Hypersonic Vehicles** a relatively novel population based technique, the Fireworks Algorithm (FWA), with additional hybrid elements for the control of a highly nonlinear problem with uncertainty elements: the high order sliding mode control of hypersonic vehicles. The FWA is combined with some elements of the genetic algorithm, and the hybrid approach applied here shows close relationship with the philosophy of the Bacterial Evolutionary Algorithm where better and worse solutions are always kept and combined for ensuring higher diversity. The main challenge here is to achieve a satisfactory tracking control performance under uncertain elements in the system and in the environment. In the study, first, the complex relation between design parameters and the cost function that evaluates the likelihood of system instability and violation of design requirements is modeled via stochastic robustness analysis. The proposed method is applied for parameter optimization and with the parameters thus obtained the efficiency of the proposed hybrid FWA-based optimization method is demonstrated in the search of the optimal HV controller, in which the proposed method exhibits a better performance when compared with other algorithms.

An interesting approach to solve the problem of finding suitable parameters of complex models that satisfy specific requirements is presented in the paper by B. G. Kang et al., being entitled **Simulation-Based Optimization on the System-of-Systems Model via Model Transformation and Genetic Algorithm: A Case Study of Network-Centric Warfare**. The authors propose a simulation-based optimization approach for modeling and make use of the System of Systems (SoS) model. The approach addresses two difficult yet important aspects: the long simulation times and a need for extensive simulation-based analysis of different scenarios. The authors attempt to solve these problems via transforming an SoS model into a neural network and further applying genetic algorithms for finding optimal values of the model parameters.

Location inventory problems are addressed in the next two works coauthored by the same group of researchers. The first work, **An Improved Differential Evolution Algorithm for a Multicommodity Location-Inventory Problem with False Failure Returns**, by C. Li et al. focuses on a multicommodity location-inventory problem. The authors propose a mixed-integer nonlinear programming-based model for studying a forward-reverse logistics network. The model allows for minimizing a total cost of false failure returns. It is solved using a proposed new population type heuristic methodology.

A different although related task of minimizing a total cost in the case of location-inventory-routing problem is considered in **A Nonlinear Integer Programming Model for Integrated Location, Inventory, and Routing Decisions in a Closed-Loop Supply Chain**. H. Guo et al. formulate the problem as a nonlinear integer programming model. They use it for optimization of such issues as facility location, inventory control, and vehicle routing. Further, they solve the model using an algorithm that combines simulated annealing with adaptive genetic algorithm.

Processes of decision-making in industrial settings are covered in the paper entitled **A Binary Cuckoo Search Big Data Algorithm Applied to Large-Scale Crew Scheduling Problems** by J. Garcia et al.. The authors apply meta-heuristic techniques to a scenario involving big data and the Internet of Things. In particular, they propose a Cuckoo Search Binary algorithm that uses the map-reduce programming paradigm of the Apache Spark tool. They apply it to a crew scheduling problem. Such aspects as convergence times and conditions for obtaining acceptable results are investigated. The form of decision making as a selection of project portfolio may be especially critical in the context of software industry.

J. Xiao et al., in **An Improved MOEA/D Based on Reference Distance for Software Project Portfolio Optimization**, treat multiobjective evolutionary algorithms as an optimization tool to solve selection tasks. They obtain a Pareto-optimal front as the solution. The paper introduces and describes an improved version of the classic multiobjective algorithm. The modification means incorporating a reference distance. The algorithm is used with two, three, and four objectives. An extensive investigation and comparison with existing multiobjective algorithms are provided.

J. Lee et al. present the paper **Effective Evolutionary Multilabel Feature Selection under a Budget Constraint**, which improves conventional methods that frequently violate budget constraints or result in inefficient searches due to ineffective exploration of some important features. The proposed method employs a novel exploration operation to enhance the search capabilities of a traditional genetic search, resulting in improved multilabel classification. Moreover, an empirical study is introduced, which is based on 20 real-world datasets showing evidence for the advantageous features of the proposed method.

The paper **Computational Analysis of Complex Population Dynamical Model with Arbitrary Order**, written by F. Haq et al., considers the approximation of the solution of a fractional order biological population model. From the Laplace Adomian decomposition method (LADM), which uses the Caputo sense for the fractional derivative, the authors construct a base function and provide a deformation equation of higher order in a simple equation. The considered scheme provides a solution in the form of rapidly convergent infinite series. The paper also includes some examples to check the efficiency of the method. As a consequence, the authors show that LADM is an efficient and accurate method for solving such type of nonlinear problems.

G. Cabrera-Guerrero et al. present a study entitled **Parameter Tuning for Local-Search-Based Matheuristic Methods**, which focuses on the parameter α that determines

the size of the subproblem that is generated by the heuristic method and is solved by the exact method. They show how meta-heuristic performance varies as this parameter is modified. This paper considers a well-known NP-hard combinatorial optimisation problem, namely, the capacitated facility location problem as the experimental basis. Based on the obtained results, they discuss the effects of adjusting the size of subproblems that are generated when using meta-heuristics methods, such as the one considered in the paper. The paper also aims at studying the impact of parameter tuning on the performance of matheuristic methods.

The paper by Z. W. Geem et al. entitled **Improved Optimization for Wastewater Treatment and Reuse System using Computational Intelligence** proposes an optimization method applying computational intelligence for reuse system. River water pollution by wastewater can cause a significant negative impact on the aquatic sustainability. Hence, accurate modeling of this complicated system and its cost-effective treatment and the reuse decision is very important, because this optimization process is related to economic expenditure, societal health, and environmental deterioration. In order to optimize this complex system, three treatment or reuse options are considered, namely, micro-screening filtration, nitrification, and fertilization-oriented irrigation, on top of two already existing options, namely, settling and biological oxidation. The objective of the environmental optimization is to minimize the economic expenditure of life cycle costs while satisfying the public health standards in terms of groundwater quality and the environmental standards in terms of river water quality. The study improves the existing optimization model by pinpointing the critical deficit, pointing at the dissolved oxygen sag curve by using analytic differentiation. The proposed formulation considers more practical constraints such as maximal size of the irrigation area and the minimal amount of the filtration treatment process. The results are obtained by using an evolutionary type algorithm, named parameter-setting-free harmony search algorithm, showing that the proposed model finds optimal solutions successfully, while eliminating the critical deficit point in the previous approaches.

The paper by X. Lv et al. entitled **An Improved Test Selection Optimization Model Based on Fault Ambiguity Group Isolation and Chaotic Discrete PSO** proposes an improved test selection model for optimization. Sensor data-based test selection optimization is the basis for designing test procedures so that they ensure that the system is tested under the constraint of the conventional indices, such as fault detection rate (FDR) and fault isolation rate (FIR). From the perspective of the equipment maintenance support, the ambiguity isolation has a significant effect on the result of the test selection. In this study, an improved test selection optimization model is proposed by considering the ambiguity degree of fault isolation. In the new model, the fault test dependency matrix is adapted to model the correlation between the system fault and the test group. The objective function of the proposed model is minimizing the test cost with the constraint of FDR and FIR. The improved chaotic discrete Particle Swarm Optimization (PSO) algorithm is applied for solving the improved test

selection optimization model. The new model is more consistent with complicated real life engineering systems and the experimental results verify the effectiveness of the proposed method.

The next article authored by L. Liu et al. is entitled **Legendre Cooperative PSO Strategies for Trajectory Optimization** and it introduces some novel strategies for trajectory optimization. PSO is a population based stochastic optimization technique which performs well in a smooth search space. However, in the case of the trajectory optimization problem with arbitrary final time and multiple control variables, the smoothness of variables cannot be satisfied, since here linear interpolation is widely used. In the paper, a novel Legendre Cooperative PSO (LCPSO) is proposed by proposing the use of Legendre orthogonal polynomials instead of linear interpolation. An additional control variable is introduced which transcribes the original optimal problem with arbitrary final time to a fixed one. Then, a practical fast one-dimensional interval search algorithm is designed to optimize this additional control variable. In order to improve the convergence and prevent explosion of the LCPSO, a theorem on how to determine the boundaries of the coefficient of polynomials is proved. Finally, in the numerical simulations, compared with the ordinary PSO and other more classical population based optimization algorithms, namely, GA and DE, it is evidenced that the proposed LCPSO has lower dimensionality, faster speed of convergence, and higher accuracy, while providing smoother control variables.

The paper by A. O. Belousov and T. R. Gazizov entitled **Systematic Approach to Optimization for Protection against Intentional Ultrashort Pulses Based on Multiconductor Modal Filters** proposes a new approach of optimization for protection against intentional ultrashort pulses. The problem of protecting radio electronic equipment from ultrashort pulses is of utmost importance nowadays since conductive interference poses the biggest danger to its proper functioning. The article considers the issue of protecting equipment by means of modal filters (MF) and analyzes the structures of multiconductor microstrip MFs. They present the results of a complex study of the possibility to conduct the optimization (both separate and simultaneous) of a multiconductor MF by different criteria and the formulation of the basic (electrical) optimization criteria for MF. They formulate the amplitude and time criteria for optimizing an MF with an arbitrary number of conductors in an analytical form and thus obtain a general multicriteria objective function for optimizing an MF by different criteria. As a result, they formulate a hybrid model consisting of heuristic search and GA.

The study by Z. Nasar and S. W. Jaffry entitled **Trust-Based Situation Awareness: Comparative Analysis of Agent-Based and Population-Based Modeling** provides an insight in the field of trust based communication in the multi-agent simulation modeling. In the paper, the authors present a comparison study on two popularly used approaches of modeling multiagent systems under situations with homogeneous and heterogeneous populations. The two approaches used in the study are the Agent Based Modeling (ABM) and Population Based Modeling (PBM) methods. The paper

also provides results on showing the sensitivity of the trust analysis, especially in the case of heterogeneous systems.

The next two papers represent somewhat different approaches, as actual biological and biochemical systems are investigated by evolutionary-population type methods. In the second of the two, a fuzzy modeling method is also used, so it may be considered as the bridging article towards the next type of CI methods.

In the study **Computational Analysis of Complex Population Dynamical Model with Arbitrary Order**, F. Haq et al. tackle the solution of highly nonlinear problems, by applying a population dynamical model, where, however, some classic mathematical tools, such as the Laplace Adomian decomposition method (LADM), are applied. The importance of fractional differential equations in the modeling of high complexity systems is without doubt, and by applying traditional mathematical approaches, such equations may not be solved at all, or if yes, in a very time consuming way. Numerical calculations may be also very resource demanding, and here, the accuracy of the solution is questionable. The authors briefly survey the earlier attempts for solving this class of problem and then introduce the necessary mathematical tool box for being able to apply Laplace transform on fractional differential equations, such as the Caputo-derivative and the Riemann-Liouville fractional order integral operator. The applicability of the proposed method is demonstrated on a fractional order biological model. They show that their approach is more efficient, especially, converging better than previous approaches. Even though this method may be considered as a traditional technique, however, its biological application makes it interesting for this special issue and points towards further open questions in the field of hybrid CI models and algorithms, which may be answered in the future by combining this novel method with population based or other optimization approaches, so we decided to include the paper among the selected ones.

Biological systems are one of the most complex ones. Therefore, an application of computational intelligence techniques for their modeling in order to better understand their behavior and interrelations may be interesting and promising. An example for that is presented in the paper **Simulations of Higher-Order Protein Organizations Using a Fuzzy Framework** by B. Tüü-Szabó et al.. The authors address a challenging task of gaining insight into organizational principles of higher-order structures, such as proteins and nucleic acids. They propose a fuzzy mathematical framework to model and investigate interaction patterns and the multiplicity of conformational states in protein assemblies. Analysis of the model allows the authors to draw some novel conclusions about associations of biological polymers, and polymer formation in particular.

Having entered the fuzzy systems area, the next papers will be discussed, which all apply fuzzy models, although, in some cases, special, extended fuzzy concepts.

P. Baranyi discusses the need for certain advantageous features, such as rank/complexity reduction, trade-offs between complexity and accuracy, and a manipulation power representative of the Tensor Product (TP) form in the

article **Extension of the Multi-TP Model Transformation to Functions with Different Numbers of Variables**. In this regard, the paper presents a novel TP model transformation based concept in the Takagi-Sugeno (TS) type fuzzy modeling and control. The latest extensions of the TP model transformation, called the multi- and generalized TP model transformations, are applicable to a set of functions where the dimensionality of the outputs of the functions may differ; however, there is a strict limitation on the dimensionality of their respective inputs, and namely, they must have the same value. This paper proposes a new, extended version of the TP model transformation that is applicable to a set of functions where both the input and output dimensionalities of the functions may differ. This makes it possible to transform complete multicomponent systems to TS fuzzy models along with the above-mentioned advantages in the complexity reduction and manipulation of the model representation.

J. Wang et al. study the topic of hesitant fuzzy systems in the paper **Some Hesitant Fuzzy Linguistic Muirhead Means with Their Application to Multiattribute Group Decision-Making**. Specifically, they extend the Muirhead mean (MM), which is a useful aggregation technology that is able to consider the interrelationship among all aggregated arguments, to a hesitant fuzzy linguistic environment. As a consequence, several new hesitant fuzzy linguistic aggregation operators may be introduced. These operators reflect the correlations among all the hesitant fuzzy linguistic elements. Furthermore, they propose a novel approach to multiattribute group decision-making (MAGDM) in a hesitant fuzzy linguistic context, based on the proposed operators. The paper concludes with a numerical experiment to demonstrate the validity of the proposed methodology and a comparison with other related methods.

In the paper by F. Lilik et al. entitled **Improved Method for Predicting the Performance of the Physical Links in Telecommunications Access Networks**, the authors propose a novel method of combining fuzzy inference and fuzzy rule interpolation techniques with wavelet transform, in order to predict the data transmission rate of a telecommunication network. Given the importance of such networks used for the transmission of not only voice but also data, the ability of accurate modeling and simulation in order to predict the quality of the network is very important. The analysis of noise and signal loss function is an important area to investigate. The authors provide results on applying wavelet analysis and use fuzzy rule inference by fuzzy rule interpolation for generating verbal rules. The method offers a robust prediction model with essential reduction of the necessary amount of measurements, compared with existing best practice.

A combination of the fuzzy and the neural network approach is applied in the next study.

S. Sotirov et al. present an intuitionistic fuzzy sets based method in **A Hybrid Approach for Modular Neural Network Design using Intercriteria Analysis and Intuitionistic Fuzzy Logic**, in order to remove a part of the inputs and, hence, the neurons. This method also leads to the decrease of the error between the desired goal value and the real value

obtained on the output of a Modular Neural Network (MNN). This type of neural networks combines several simple neural models for reducing the complexity of a solution of a complex problem, which can be used, e.g., for object recognition and identification. Usually, the inputs of the MNN can be fed with independent data. However, there are certain limits in the use of MNN, and the number of the neurons is one of the major parameters during the implementation of it. The proposed method allows increasing the velocity in the learning process, a much desired effect. Furthermore, this method based on intercriteria analysis and intuitionistic fuzzy logic can also be used with good results for assessing the independence of data.

In the article **The Intuitionistic Fuzzy Linguistic Cosine Similarity Measure and Its Application in Pattern Recognition** by D. Liu et al., the authors propose the use of cosine similarity measures to enhance the precision of the intuitionistic fuzzy linguistic sets (IFLS) and interval-valued intuitionistic fuzzy linguistic sets (IVIFLSs) in a modeling problem. IFLSs were introduced in order to handle applications where the decision needs to be performed under (partially) nondeterministic conditions in the linguistic evaluation and where it is difficult to handle the problem by using only membership degrees of linguistic terms. The authors' claim is that, at the time of the publication, there has been no work yet done on the study of using cosine similarity measures on IFLSs. The paper thus provides studies to fill this gap.

In the third block the papers applying neural network (NN) techniques are briefly introduced, although the first one shows certain characteristics of evolutionary approaches as well.

A. B. Csapo proposes a neural network-inspired formulation of the Spiral Discovery Method (SDN) in the work **The Spiral Discovery Network as an Automated General-Purpose Optimization Tool**, a cognitive artifact that was originally designed for user-guided interactive search in high-dimensional nonlinear parameter spaces. The SDN proposed in this paper extends that model by complementing its autoregressive update profile with a periodically activated hyperparameter update function, which serves for modeling, and replacing, the interventions that were previously carried out by the user. The SDN can be interpreted as an adaptive search algorithm that shows commonalities with supervised neural networks, evolutionary methods, and other optimization methods, but is also different in the sense that it does not directly rely on gradient-based feedback or on biologically inspired concepts such as genes or populations. The applicability of the approach is demonstrated through an example search space that is characterized by misleading gradient information and a narrow global minimum.

The next papers are purely NN approaches. An NN often allows obtaining relevant dynamic information of unknown nonlinear systems. In this framework, E. Irigoyen et al. present the paper **About Extracting Dynamic Information of Unknown Complex Systems by Neural Networks**. Based on the assumption that the dynamic behavior is excessively challenging to obtain an accurate mathematical model,

considering Multilayer Perceptron (MLP) they present a system representation using a model formulated with state variables, which can be exported to an NN structure. The equilibrium states are studied by calculating the Jacobian matrix of the system through the NN model and different examples are analyzed.

In the article by M. I. Dieste-Velasco et al. entitled **Regression and ANN Models for Electronic Circuit Design**, the authors investigate the use of NNs for modeling the design of electronic circuits. The paper provides a comparison with the commonly used approach that is a simple regression analysis in order to construct the model. The analysis of electronic circuits with various input parameters can be rather complex. The paper demonstrates that a straightforward NN can provide a good analytic model by helping understand the behavior of current and voltage in electronic circuits.

Eventually, a nowadays rare CI approach, a chaos based model is proposed in the next work.

In **Experimental Verification of Optimized Multiscroll Chaotic Oscillators Based on Irregular Saturated Functions**, J. M. Muñoz-Pacheco et al. discuss an approach where they apply multiscroll chaotic attractors generated by irregular saturated nonlinear functions. These functions are designed in an irregular way by modifying their parameters, such as slopes, delays between slopes, and breakpoints, and then the positive Lyapunov exponent (LE) is optimized using the differential evolution algorithm to obtain chaotic attractors with 2 to 5 scrolls. The resulting chaotic attractors present more complex dynamics when different patterns of irregular saturated nonlinear functions are considered. The optimized chaotic oscillators have been physically implemented with help of an analog discrete circuit to validate the use of proposed irregular saturated functions. Experimental results achieved are consistent with two different types of simulators. The authors compare the results with other approaches using various evolutionary optimization algorithms and conclude that the new chaos-based technique produces better LE values than the other two used as reference.

Finally, a method applying subjective probabilistic approach closes the series of studies on CI applications for complex problems.

Food is an important resource all over the world, especially in areas with growing population. In the paper by Y. E. Shao and J.-T. Dai on **Integrated Feature Selection of ARIMA with Computational Intelligence Approaches for Food Crop Price Prediction**, the authors investigate the ways of using computational intelligence to provide an accurate food crop price prediction. In the paper, they provide investigation of using single models and integrated models for predicting food crop prices such as rice, wheat, and corn. In the integrated models, the authors have used the ARIMA as a feature selection method.

We hope that the Reader will enjoy this selected collection of papers, and the novel scientific ideas in them will be the starting points of new research, and will, maybe, trigger entirely new ideas, and thus will lead to more efficient solutions of highly complex problems in both basic and applied fields.

Conflicts of Interest

The editors declare that they have no conflicts of interest regarding the publication of this special issue.

*Laszlo T. Koczy
Jesus Medina
Marek Reformat
Kok Wai Wong
Jin Hee Yoon*

Research Article

Simulations of Higher-Order Protein Organizations Using a Fuzzy Framework

B. Tüü-Szabó ¹, **L. T. Kóczy** ^{1,2} and **M. Fuxreiter** ³

¹Department of Information Technology, Széchenyi István University, Győr, Hungary

²Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary

³MTA-DE Laboratory of Protein Dynamics, University of Debrecen, Hungary

Correspondence should be addressed to L. T. Kóczy; koczy@tmit.bme.hu and M. Fuxreiter; fmoni@med.unideb.hu

Received 28 December 2017; Revised 29 August 2018; Accepted 12 September 2018; Published 2 December 2018

Academic Editor: Dimitri Volchenkov

Copyright © 2018 B. Tüü-Szabó 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.

Spatiotemporal regulation of the biochemical information is often linked to supramolecular organizations proteins and nucleic acids, the driving forces of which have yet to be elucidated. Although the critical role of multivalency in phase transition has been recognized, the organization principles of higher-order structures need to be understood. Here, we present a fuzzy mathematical framework to handle the heterogeneity of interactions patterns and the resultant multiplicity of conformational states in protein assemblies. In this model, redundant binding motifs can establish simultaneous and partial interactions with multiple targets. We demonstrate that these multivalent, weak contacts facilitate polymer formation, while recapitulating the observed valency-dependence. In addition, the impact of linker dynamics and motif binding affinity, as well as the interplay between the two effects was studied. Our results support that fuzziness is a critical factor in driving higher-order protein organizations, and this could be used as a general framework to simulate different kinds of supramolecular assemblies.

1. Introduction

Proteins can form a wide variety of assemblies, in terms of composition, size, and dynamics. In addition to simple binary, ternary complexes, and middle-size oligomers, proteins may also assemble into higher-order organizations. These supramolecular assemblies are implicated in different biological processes ranging from normal physiology to disease [1, 2]. For example, to minimize signaling noise for low-affinity effectors, signaling complexes frequently increase the local concentration of binding sites via higher-order protein assembly [3]. Recent discoveries revealed that supramolecular organizations of proteins and nucleic acids can generate functional cellular compartments [4, 5], which lack a membrane boundary [6]. Such membraneless organelles appear at various points on the biological landscape, for example, can serve as biomolecular storages upon stress and bioreactors to accelerate chemical reactions as well as signaling devices, whose assembly/disassembly is regulated by a variety of pathways [7–9]. Seminal works by Brangwynne, Hyman, and Parker labs revealed that these organelles are

created by a process of liquid-liquid demixing, once the component concentration exceeds the saturation limit [10, 11]. This process, which was termed as a phase transition, could be described by the Flory–Huggins theory [12].

Higher-order protein organizations exhibit a wide spectrum of states with distinct dynamics. Prions/amyloids are stabilized by β -zipper, resulting in static and solid-like inheritable entities [13]. Signalosomes, such as inflammasomes or necrosomes could resemble prion-like stable structures [14] or be dynamic, for example, the autophagosome [15]. Ribonucleoproteins (RNP) generate dynamic granules or liquid-like droplets [16]. Nuclear pore complexes (NPCs) are somewhat more stable and form hydrogels [17]. Intriguingly, the very same protein could be organized into different higher-order states with distinct dynamics. Pathological mutations may induce conversion of the material state, for example, liquid-like droplets to solid fibrils. In the case of the hnRNPA and Fus protein, familial mutations appear in Amyotrophic lateral sclerosis (ALS) [1, 2]. Interestingly, pathological mutations often affect intrinsically disordered linker regions [18] and not the interacting motifs themselves,

suggesting the importance of conformational dynamics in organizing higher-order structures [19].

These experimental observations are in agreement with the recently proposed framework for higher-order protein organizations [20]. This model suggests that the material state of higher-order structures could be decomposed into three factors. First, low-affinity elements/motifs such as cation- π , π - π , aromatic hydrogens bonds [21, 22] mediate weak contacts with fast off-rates. Second, multivalency of binding elements increases the number of microstates in the bound state [23]. Third, conformational flexibility/dynamics or disorder must be retained to enable different topologies and reduce the entropic penalty of binding [16]. Along these lines, NMR data indicates similar conformational heterogeneity of Fus in its free and bound states [16]. Polymer physics approaches recapitulated the effect of multivalency of interacting motifs [23]. Monte Carlo simulations shed light on linker solvation [24, 25]. However, the problem of degenerate interaction patterns by weak affinity motifs [22] and how the resultant heterogeneous conformations impact higher-order protein organizations have not been addressed so far.

Our model considers the supramolecular assembly as a fuzzy complex [26–32]. These protein assemblies are characterized by structural multiplicity or dynamical disorder in their bound states, and the conformationally heterogeneous region is demonstrated to have a considerable impact on the biological function. This phenomenon is referred to as “fuzziness” in proteins, which concept has been supported by experimental evidence on a wide range of examples [33]. Fuzziness originates in transient and ambiguous interactions, which lead to redundant contact patterns in protein complexes [27, 34]. Although the view on degenerate alternative contacts in specific complexes contradicts to the traditional concept on specific molecular recognition, it is corroborated by a wide range of experimental data [35, 36] (and references therein). Furthermore, a variety of regulatory mechanisms are linked to protein fuzziness [29], which might also contribute to the organization of higher-order assemblies [20].

Fuzziness is known as a mathematical concept, where the membership in given sets is described by a function, varying between [0,1] instead of a single [0 or 1] value. Fuzziness has been derived from the seminal work of Zadeh [37] and has been implicated in the electronic control of ~3000 artificially intelligent devices [38]. We have developed a simulation method based on the mathematical concept of fuzziness to describe how weak degenerate motifs connected by flexible linkers can be organized into higher-order polymers. In this model, a single binding site can simultaneously interact with multiple binding motifs to different extents. We show that these partial and multiple contacts facilitate polymer formation as compared to the one-to-one binding models. Independently of the fuzzy description, the algorithm allows to study the influence of linker dynamics, motif affinity, and their interplay on molecular associations. Our results support the previous proposal [20] that fuzziness is a critical factor in driving higher-order protein organizations.

2. Methods

2.1. Model System. The simulations were run on 100 hypothetically identical proteins, which were placed into 64 periodic simulation boxes. All units are given in residues. The separate proteins can associate, the resultant polymers can dissociate, and both can diffuse with probabilities given below.

2.2. Protein. The model is a hypothetically biological chain, which is composed of N residues (depending on the valence, Figure 1). Each residue is characterized by two values: binding affinity and dynamics. Binding affinity characterizes the interaction preference, and this value could be derived from experimental studies or theoretical estimates of binding free energies. Here, dynamics refers to a conformational exchange in the bound state. That is, upon molecular interaction, to what extent flexibility or dynamical disorder can be retained in the assembly. If this value is high, the residue can interconvert between multiple states, generating multiple interaction patterns and heterogeneous conformations. If $D=1$, linkers preserve their conformational heterogeneity similarly to their unbound state, while in case of $D=0$, the linkers collapse and become rigid in the assembly. The value for dynamics could be derived from bioinformatics studies, as it can be predicted based on the protein sequence or from NMR measurements. In this work, the values for binding affinity and dynamics are hypothetical and not derived from observed or computed data on specific model systems.

Values for binding affinity and dynamics vary in a [0,1] range and were kept fixed during simulation. Binding affinity >0.3 designates residues involved in binding; fuzziness values >0.3 correspond to linker residues. Binding elements (α) are defined as a continuous stretch of at least 5 residues with binding affinity >0.3 . Linkers (λ) are defined as all other segments connecting the binding elements. In case of the hypothetical protein, the length of each binding site was 7 and the length of each linker was 10 residues. The size of the molecule was varied between 34 and 102 residues.

2.3. Molecular Association/Polymer. If two molecules are linked by one or more nonzero fuzzy interactions, they are considered to form a complex. Interacting sites must be spatially close (i.e., their distance should be below a given threshold). Bound molecules move together in case of diffusion to the neighboring box. Parameters for binding (affinity, probability to associate/dissociate) are described below. Large polymers (higher-order assemblies) were defined as interconnected $m > 25$ molecules.

2.4. Simulation Box. 64 cubic simulation boxes were used, with lengths between 20 and 70 residues (units are given in residues). The molecules were placed randomly into the boxes, with no steric overlap.

2.5. Parameters. Binding preference of a binding element (α_i) is obtained as an average of the residue binding preferences.

$$S_{\alpha_i} = \frac{\sum_{k=1}^{n_i} S_k}{n_i}, \quad (1)$$

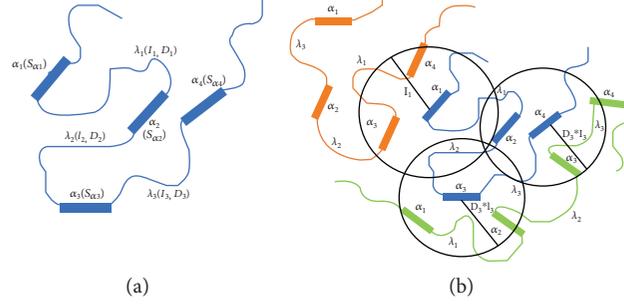


FIGURE 1: Schematic representation of the model system with one (a) and three interacting molecules (b). s_{α_i} is the binding preference of a binding element (α), which is computed as the average of the residue-based binding affinities (equation (1)). D_i is the dynamics of the linker (λ) with length l_i , which is obtained as the average of the residue-based values (equation (5)). The local concentration of the available binding sites is computed within a volume V_{l_i} , which is scaled by the linker dynamics (D_i , equation (6)).

where s_k is the binding preference of residue k and the binding element contains n_i residues.

The affinity between two binding elements is defined as the average of the binding preferences [0,1].

$$K_{\alpha_i, \alpha_j} = \frac{s_{\alpha_i} + s_{\alpha_j}}{2}, \quad (2)$$

where s_{α_i} and s_{α_j} are the binding preferences of the interacting α_i and α_j elements.

As the fuzzy framework allows one binding element to interact with multiple other elements, the number of possible binding elements available for interaction needs to be determined. All binding elements within the volume, which is defined by the neighboring linkers are considered (see below).

$$n_{\maxint, \alpha_i} = \sum_{j=1}^n \alpha_j, \quad (3)$$

where $\alpha_j \in V_{l_i}$, and n_{\maxint, α_i} is the maximum number of interaction sites around the binding element α_i . The available volume for α_i interactions is defined by the length of the longer neighboring linker (l_i), which is measured from the center of the binding element. In the default case, V_{l_i} is a spherical volume with a radius of l_i .

$$V_{l_i} = \frac{4\pi l_i^3}{3}. \quad (4)$$

This volume could be rescaled according to linker dynamics, which is defined as the average of the residue dynamics.

$$D_i = \frac{\sum_{k=1}^l d_k}{l}, \quad (5)$$

where d_k is the dynamical value of residue k , and l_i is the length of the linker.

If the linker dynamics (D_i) is 1, all the available binding sites are considered within the volume as defined in equation (4). If $D_i < 1$, the sphere radius is reduced proportionally to the linker dynamics.

$$l'_i = D_i * l_i, \quad (6)$$

where D_i is the linker dynamics and l_i is the length of the linker. l'_i is used to obtain the volume by equation (4).

2.6. Computed Quantities. The association probability linearly depends on the binding affinity (K_{α_i, α_j}) and reciprocally on the available binding sites (n_{\maxint, α_i}).

$$p_{\text{on}}^{\text{eff}} = \frac{p_{\text{on}}^{\text{int}} * (1 - p_{\text{off}}^{\text{int}}) * K_{\alpha_i, \alpha_j}}{n_{\maxint, \alpha_i} + n_{\maxint, \alpha_j}}, \quad (7)$$

where the intrinsic association probability $p_{\text{on}}^{\text{int}} = 0.6$, the intrinsic dissociation probability $p_{\text{off}}^{\text{int}} = 0.1$, similarly to the reference [23]. The association probability is compared to a random number (rnd), and if $p_{\text{on}}^{\text{eff}} \geq \text{rnd}$, the binding is realized.

Once interactions are formed in the first step, we define an occupancy value for each binding elements [0,1]. It is calculated with an algebraic sum, which is the fuzzy union (s-norm) operator.

$$\begin{aligned} \mu_{\alpha_i} &= s(K_{\alpha_i, \alpha_1}, K_{\alpha_i, \alpha_2}, \dots, K_{\alpha_i, \alpha_n}) \\ &= s(s(K_{\alpha_i, \alpha_1}, K_{\alpha_i, \alpha_2}), \dots, K_{\alpha_i, \alpha_n})s(K_{\alpha_i, \alpha_1}, K_{\alpha_i, \alpha_2}) \\ &= K_{\alpha_i, \alpha_1} + K_{\alpha_i, \alpha_2} - K_{\alpha_i, \alpha_1} * K_{\alpha_i, \alpha_2}, \end{aligned} \quad (8)$$

where n is the number of the binding elements in the polymer. The algebraic sum was chosen because of its super idempotent property $s(a,a) \geq a$. Equation (8) considers all binding sites within the polymer, but only the connected sites (where $K_{\alpha_i, \alpha_j} > 0$) are taken into account in the algebraic sum.

From the second step, the affinity of a given interaction between α_i and α_j also depends on the local concentration

of the bound binding elements. The binding affinity between two elements must be weighted by the occupancies of the neighboring binding sites.

$$K'_{\alpha_i, \alpha_j} = \frac{s_{\alpha_i} + s_{\alpha_j}}{2} * \sqrt{1 + \sum_{n=1}^{n_{\max, \alpha_i}} \mu_n + \sum_{m=1}^{n_{\max, \alpha_j}} \mu_m}, \quad (9)$$

$$K'_{\alpha_i, \alpha_j} \in [0, 1],$$

where μ_n and μ_m occupancies are summarized for all binding elements within the available volume for α_i and α_j . If the local concentration is considered from the second step, the modified affinities (K'_{α_i, α_j}) are used to determine the occupancies in equation (8).

The association probability is also modified accordingly.

$$p_{\text{on}}^{\text{eff}} = \frac{p_{\text{on}}^{\text{int}} * (1 - p_{\text{off}}^{\text{int}}) * K'_{\alpha_i, \alpha_j}}{n_{\max, \alpha_i} + n_{\max, \alpha_j}} * \frac{1}{1 + n_{\text{int}, \alpha_i} + n_{\text{int}, \alpha_j}}, \quad (10)$$

where K'_{α_i, α_j} is the modified binding affinity defined in equation (9), and n_{int, α_i} and n_{int, α_j} are the actual number of binding elements, which are bound to α_i and α_j , respectively.

The dissociation probability has an inverse relationship to the binding affinity.

$$p_{\text{off}}^{\text{eff}} = \frac{p_{\text{off}}^{\text{int}} * (1 - p_{\text{on}}^{\text{int}})}{K'_{\alpha_i, \alpha_j}}, \quad (11)$$

where the intrinsic association ($p_{\text{on}}^{\text{int}}$) and dissociation ($p_{\text{off}}^{\text{int}}$) probabilities are the same as in equation (7), and K'_{α_i, α_j} is the modified binding affinity defined in equation (9).

From the second step, the molecules could be present as individual chains, or chains organized into oligomers or larger polymers. Here, we need to define the interaction capacity (“freedom”) of the binding elements within the molecular assembly/polymer.

$$F = \sum_{i=1}^n (1 - \mu_i) \quad (12)$$

where $(1 - \mu_i)$ is the available interaction capacity of a given α_i binding element, which is summarized for all binding elements in the polymer.

Any molecule types in the system: individual molecules, oligomers, or larger polymers can diffuse to another box. Diffusion means repositioning into another box, in case of polymers molecules move together. The probability of the diffusion is defined as

$$p_{\text{diff}} = \frac{1}{6} * \frac{1}{\sqrt{\sum_{i=1}^{n_{\text{int}}} (1 - \mu_i)}}, \quad (13)$$

where the square root of F (equation (12)) is used in the denominator.

2.7. Comparison of Fuzzy and Nonfuzzy Simulations. As in the fuzzy simulations, a partial binding is also considered to be a potential interaction (depending on the association probabilities as computed by equations (7) or (10)), we ensured that the probabilities of forming large polymers in the fuzzy and nonfuzzy simulations are comparable. First, the binding affinities in the fuzzy simulation are derived from equations (2) and (9), while in the nonfuzzy simulations they are always 1. Therefore, the association probabilities (equation (10)) of the nonfuzzy simulations are higher, unless the local concentration effect is considered. Fuzzy simulations only produce large polymers >4 valencies (binding elements) without considering the local concentration. Second, the association probability (equation (10)) is inversely proportional to the current number of binding interactions, so a potential to contact simultaneously via multiple sites does not necessarily increase the probability of polymer formation. Third, the dissociation probability (equation (11)) is inversely related to the binding affinity. Therefore, a weak (partial) binding in the fuzzy model is more likely to dissociate than a full binding (with $K = 1$) in the nonfuzzy model. Overall, the fuzzy simulations without the local concentration effect are comparable to the nonfuzzy simulations. Please note that the nonfuzzy simulations in this work are not equivalent to those in the reference [23], as our simulations also account for spatial dimensions, linker dynamics, and motif affinity.

2.8. Computational Protocol. A periodic system was defined, which contained 64 boxes, with dimensions from 20 to 70 residue units (length was varied in separate simulations). The system was comprised of 100 molecules, each composed of 34 to 102 residues. The simulated molecules were placed randomly in the boxes. The simulation protocol was similar to the stochastic rule-based simulation in reference [23].

In the first simulation step, the molecules could associate according to the probabilities given in equation (7). From the second iteration step, the molecules had three options: associate (i), dissociate (ii), and diffuse to another randomly chosen neighboring cell (iii). Occupancies (equation (8)) and the interaction capacities (equation (12)) are determined in each step, and affinities were modified accordingly (equation (9)). Association probabilities (equation (10)) and dissociation probabilities (equation (11)) were also scaled by the modified affinities (equation (9)) to account for the local concentration and binding status of the available sites. Diffusion was inversely proportional to the interaction capacity, so larger polymers have less chance to move to the neighboring box.

3. Results

All previous simulations of higher-order assemblies assumed that interactions are well-defined and each site contacts only one partner at the same time (one-to-one) [23, 24]. In contrast, the fuzzy model allows multiple interactions for the same site to different extents. In principle, the potential to

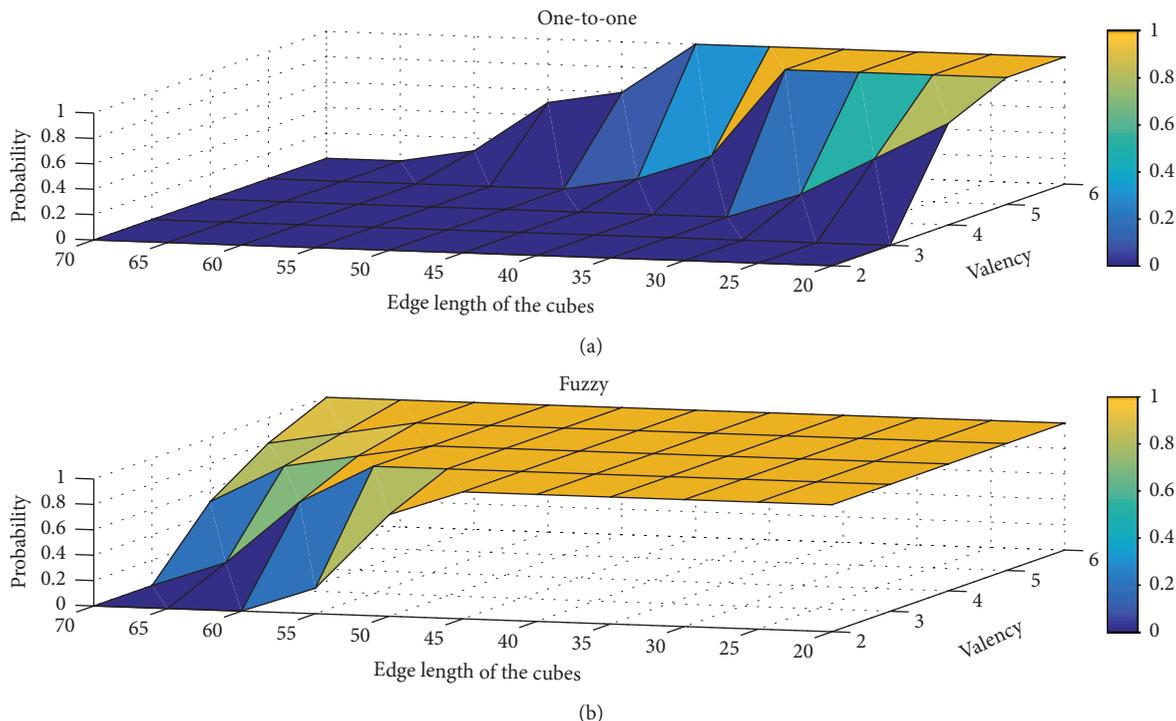


FIGURE 2: Probability of large polymers as a function of valency and length of the simulation box (in residue units) in one-to-one (a) and fuzzy (b) binding models. Valency is defined as the number of binding elements. Concentration is the number of molecules/volume (L^3). Binding element affinity = 1.0; linker dynamics = 1.0.

form multiple partial contacts can increase the probability of polymer formation. To compare fuzzy and the one-to-one (nonfuzzy) binding models, however, we eliminated the artefact that increased binding potential which causes polymer formation (Methods). First, we studied the impact of valency on the generation of higher-order assemblies. The model system contained one molecule type, the size of which has been systematically varied between 2 to 6 binding elements and linkers, and the length of which were arbitrarily defined as 7 and 10 residues, respectively (Figure 1). Concentration was modulated by varying the size of the simulation box, while the number of molecules was kept fixed. The topologies and the parameters considered for interactions are shown in Figure 1 (for explanation see Methods). Three types of stochastic movements were performed, which are similar to reference [23]: (i) association, (ii) dissociation, and (iii) diffusion to a neighboring box. The results were averaged for 10 parallel simulations (10,000 steps) for each parameter combinations.

Multivalency is considered as the major driving force of phase transition [6, 23]. The impact of valency and concentration on the probability of formation of large polymers is shown in Figure 2. Concentration is varied by changing the size of the simulation box. Both one-to-one and fuzzy simulations show a strong dependence on the number of binding elements, recapitulating previous experimental observations [23].

Fuzzy interactions seem to facilitate the formation of higher-order molecular associations (Figure 2). The same number of molecules with the same valency generate polymers in an order of magnitude larger simulation box in

the fuzzy simulations than in the nonfuzzy model. This suggests that weak partial interactions with multiple partners may enhance assembly at lower valency. This is consistent with recent experimental data on highly dynamical interacting partners [39]. Along these lines, in simulations, which were conducted using lower binding element affinities (0.35) and linker dynamics (0.35), no higher-order oligomers ($m > 25$) were observed in the one-to-one simulations, while polymers were formed in the case of > 4 binding elements in the fuzzy model (Figure 3).

We also studied the impact of binding affinity on polymerization. Intuitively, higher affinity between the interacting elements increases the association probability (equation 10). In accord, in both fuzzy and nonfuzzy models, polymerization takes place at lower valency at higher binding affinity (Figure 4). Using the same simulation conditions, where no polymerization was observed in the nonfuzzy model (Figure 3), we have observed higher-order assembly at increased affinity. In the fuzzy model, polymerization occurs at lower affinity (Figure 4(b)) as compared to the one-to-one binding, illustrating that partial and heterogeneous contacts may compensate for weaker interactions [22]. Obviously, above a certain limit, increasing affinity and valency may induce the formation of aggregates or amyloid structures and not dynamical assemblies.

In the fuzzy model, we assume that the local concentration of the binding elements also influences affinity, which is in agreement with earlier theoretical [40] and experimental [41] results. This effect was taken into account via equation (9), and the modified affinities were incorporated into

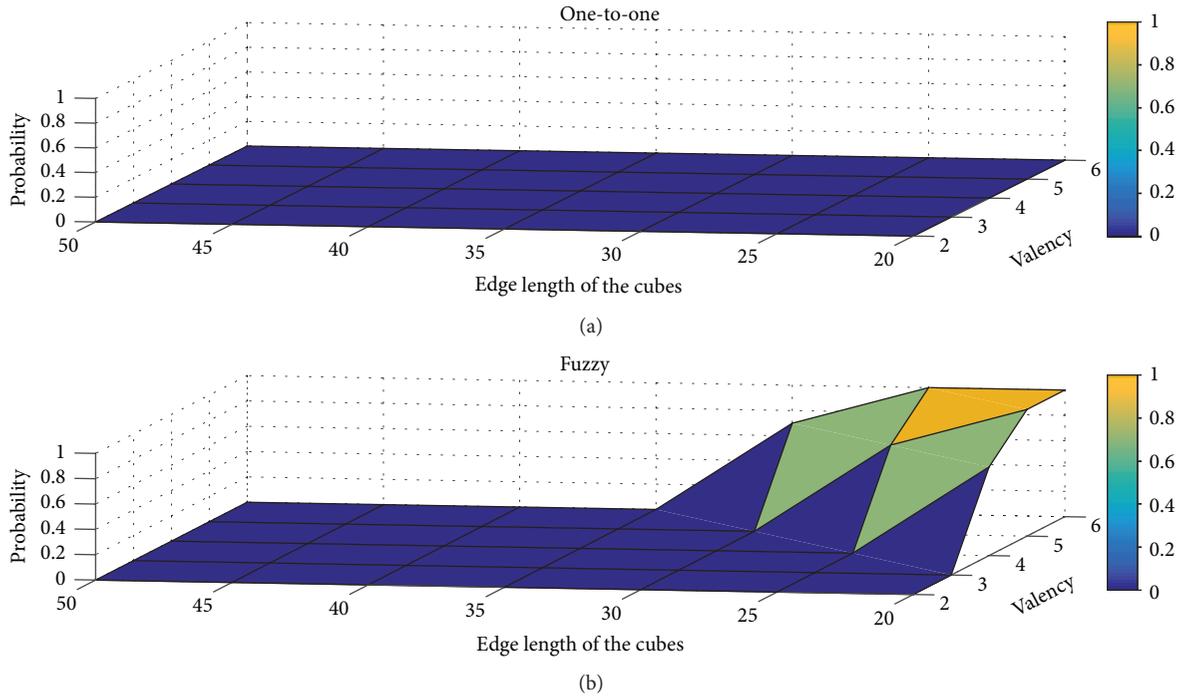


FIGURE 3: Probability of large polymers as a function of valency and length of the simulation box (in residue units) in one-to-one (a) and fuzzy (b) binding models. Valency is defined as the number of binding elements. Concentration is the number of molecules/volume (L^3). Binding element affinity = 0.35; linker dynamics = 0.35.

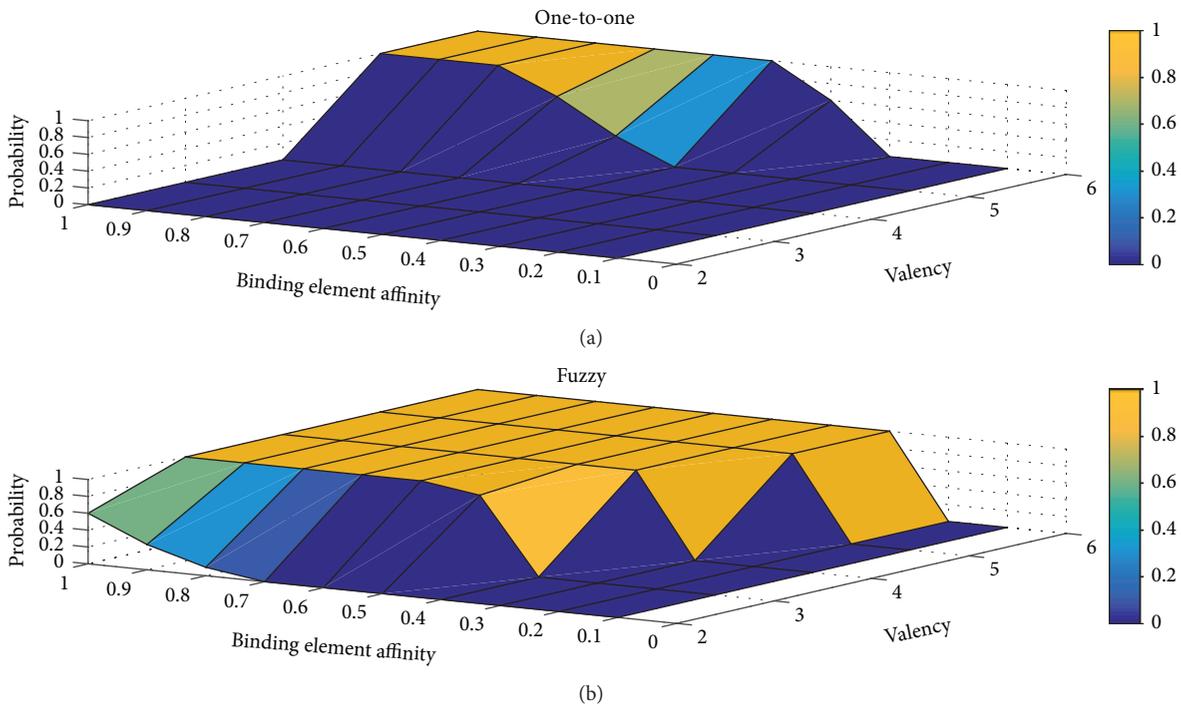


FIGURE 4: Probability of large polymers as a function of valency and binding element affinity in one-to-one (a) and fuzzy (b) binding models. Valency is defined as the number of binding elements, which affinity (s_{α_i}) is computed as the average of the residue-based values (equation (1)). Linker dynamics = 0.35; box length = 20.

the association (equation (10)) and dissociation probabilities (equation (11)). Considering the local concentration of the binding sites generates polymers at lower valency

(Figure 5). This could be one of the reasons why fuzzy simulations produce higher-order states more frequently (see Figures 5(b) and 6(a)).

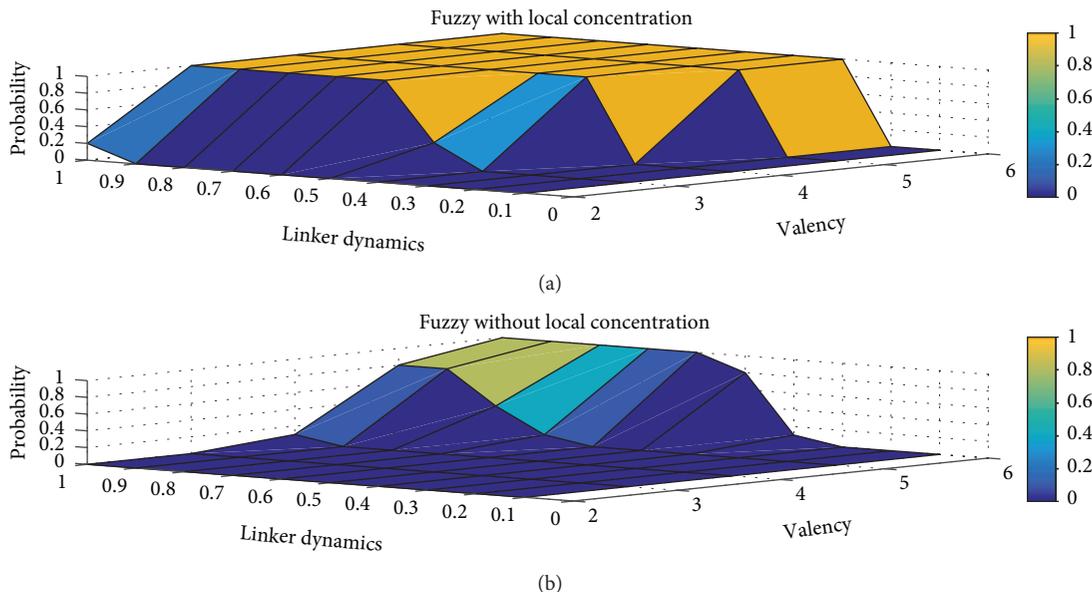


FIGURE 5: Probability of large polymers as a function of valency and binding affinity in the fuzzy binding model with (a) and without (b) considering the local concentration effect. The local concentration is computed by equation (9). Linker dynamics = 0.35; box length = 20.

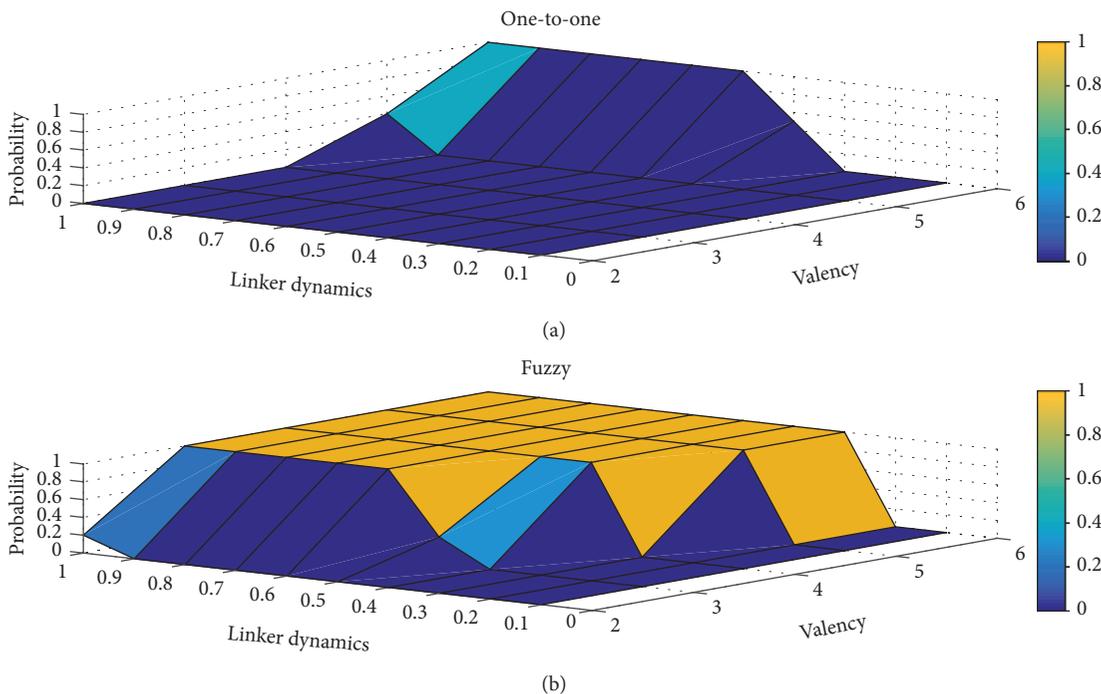


FIGURE 6: Probability of large polymers as a function of valency and linker dynamics in the one-to-one (a) fuzzy (b) binding models. Linker dynamics is computed as the average of the residue-based values (equation (5)). Binding element affinity = 0.35; box length = 20.

Intrinsically disordered regions play important roles in organizing higher-order structures [18, 42]. These proteins segments, while lacking a well-defined structure, exhibit enhanced plasticity that enables a large number of contact combinations [20, 24]. However, intrinsically disordered regions are not infinitely flexible; they exhibit a given range of dynamics, which certainly affects the available topologies.

In our fuzzy model, we take this effect into account via equation (6), which modifies the affinities and association/dissociation probabilities. Systematic increase of linker dynamics increases the probability of polymer formation (Figure 6). In agreement with experimental studies [16, 39], these results reflect that linker dynamics is a critical element of higher-order assembly, which is independent of motif

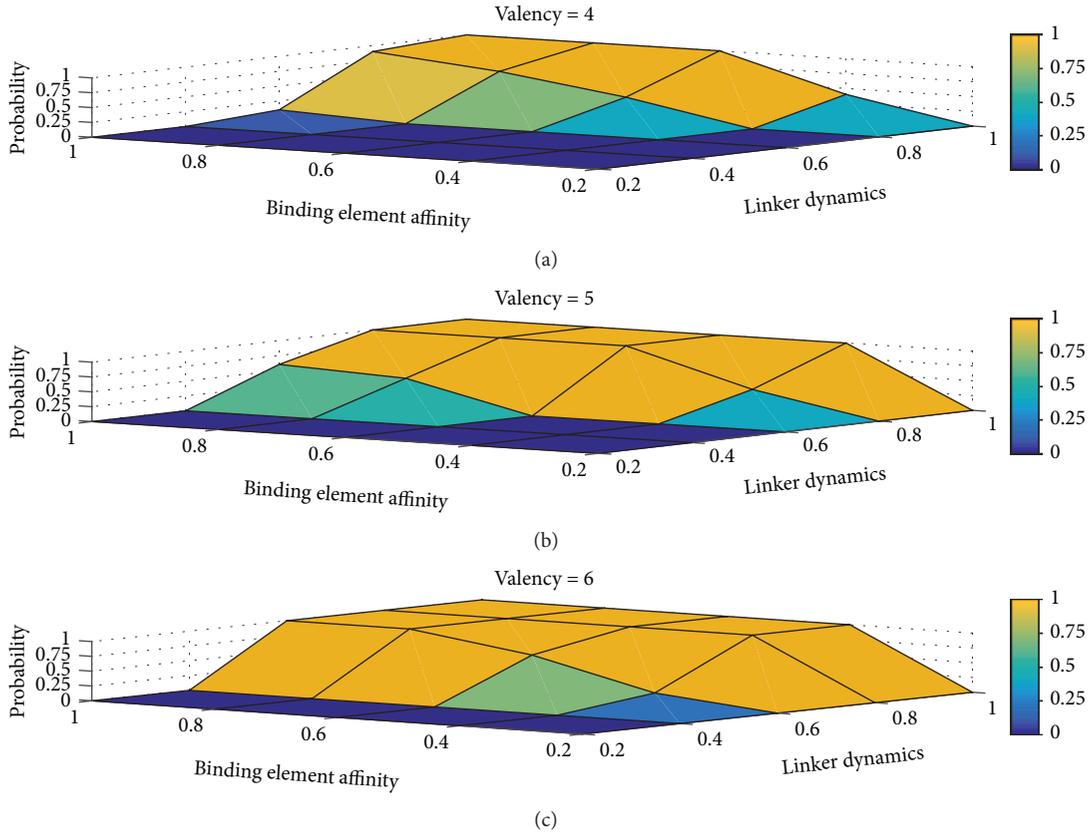


FIGURE 7: Probability of large polymers as a function of binding element affinity and linker dynamics in the fuzzy binding model for valency $n = 4$, $n = 5$, and $n = 6$.

affinity. We might hypothesize that sticky chains, with low number of valencies, are also capable to generate higher-order structures [39].

In the nonfuzzy binding model, the impact of linker dynamics is comparable to that of increasing interaction affinity (Figure 4), illustrating that either stronger or more heterogeneous binding can promote assembly formation. Fuzzy simulations are more sensitive to linker dynamics (Figure 6), as it affects the number of possible interaction patterns, which could be generated at a given time point. This observation underscores that degenerated contacts and the resultant conformational heterogeneity are important driving forces of higher-order assembly.

In the fuzzy model, the interplay between the motif affinity and linker dynamics was studied systematically using models with different valencies (Figure 7). For any combinations of binding affinity and linker dynamics, the critical role of valency is observed.

In addition to the effect of valency, interaction affinity and linker dynamics act in synergy to promote polymerization. Lower affinity elements with more flexible linkers as well as high-affinity elements with less dynamical linkers can produce higher-order assembly. Obviously, these two scenarios are expected to result in distinct supramolecular assemblies along the dynamical continuum [20]. This is an important issue for pathological mutations, which studies are currently ongoing in our laboratory.

4. Discussion

Protein function is usually interpreted within the deterministic framework of the classical structure-function paradigm. This relationship establishes a connection between a well-defined three-dimensional organization of amino acid residues and the biological activity of the resultant conformer. The classical description also involves the assumption that the intra- or intermolecular interactions generate a well-defined pattern. Increasing experimental evidence contradicts this simple picture and demonstrates that biological function may require conformation and interaction heterogeneity [32, 33]. Sequences of proteins composing membraneless organelles, for example, are enriched in redundant/degenerate motifs [22], which appear to contact in multiple ways resulting in a heterogeneous assembly [16]. Indeed, structural and interaction heterogeneity is an intrinsic feature of higher-order protein assemblies, ranging from static to highly dynamical structures [20].

Developing computational approaches to describe heterogeneous systems is a challenge. Until now, a one-to-one binding model has been employed in both coarse-grained and lattice simulations [24], which could not account for the effect of heterogeneity, resulted by multiple alternative configurations. A fuzzy mathematical framework allows coexisting alternative structures or interaction patterns in the system, which are realized to different extents.

Within the fuzzy model, a binding element may interact with multiple partners simultaneously, and the contribution to alternative states are expressed via membership functions. The membership of a binding element varies in each configuration, and the system remains heterogeneous throughout the trajectory.

Here, we applied the fuzzy framework to a hypothetical polymer characterized by binding affinity and dynamics. The simulations recapitulate the observation that multivalency is a prerequisite for phase transition [23]. As compared to the one-to-one (nonfuzzy) binding model, the fuzzy simulations predict a lower phase boundary (Figures 2 and 3). This illustrates that weak partial interactions lead to degenerate patterns favor assembly. The partial contacts increase the probability of productive interactions via local concentration effects (Figure 5). We demonstrate that—in addition to motif affinity—linker dynamics is a critical factor in driving higher-order assembly. This effect is especially pronounced in the fuzzy model (Figure 6). Systematic studies on the interplay between binding affinity and linker dynamics outline two alternative ways to promote higher-order organization of proteins (Figure 7): higher affinity for the binding motifs or increasing dynamics in the bound system. This hypothesizes that weak interaction networks in fuzzy systems are capable to organize higher-order associates.

5. Conclusion

Understanding the driving forces of higher-order protein assembly is challenging, owing to the complexity of these systems. Here, we developed a fuzzy mathematical model to simulate associations of biological polymers. In this approach, the system is described by multiple coexisting states, capturing the inherent heterogeneity of higher-order assemblies. We propose that the fuzzy model provides a general framework to study higher-order systems along the structural and dynamical continuum.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Funding

The authors received Fund nos. GINOP-2.3.2-15-2016-00044 and HAS 11015.

Acknowledgments

We thank Pal Jedlovsky Jr. for his assistance with the simulations and critical reading of the manuscript, and Tibor Sarvari for the fruitful discussions. M. F. acknowledges the financial support of the DE Excellence Program. L. T. K. is supported by NKFIH K108405 and K124055 grants.

References

- [1] A. Molliex, J. Temirov, J. Lee et al., “Phase separation by low complexity domains promotes stress granule assembly and drives pathological fibrillization,” *Cell*, vol. 163, no. 1, pp. 123–133, 2015.
- [2] A. Patel, H. O. Lee, L. Jawerth et al., “A liquid-to-solid phase transition of the ALS protein FUS accelerated by disease mutation,” *Cell*, vol. 162, no. 5, pp. 1066–1077, 2015.
- [3] M. Bienz, “Signalosome assembly by domains undergoing dynamic head-to-tail polymerization,” *Trends in Biochemical Sciences*, vol. 39, no. 10, pp. 487–495, 2014.
- [4] C. P. Brangwynne, C. R. Eckmann, D. S. Courson et al., “Germline P granules are liquid droplets that localize by controlled dissolution/condensation,” *Science*, vol. 324, no. 5935, pp. 1729–1732, 2009.
- [5] R. Parker and U. Sheth, “P bodies and the control of mRNA translation and degradation,” *Molecular Cell*, vol. 25, no. 5, pp. 635–646, 2007.
- [6] S. F. Banani, H. O. Lee, A. A. Hyman, and M. K. Rosen, “Biomolecular condensates: organizers of cellular biochemistry,” *Nature Reviews Molecular Cell Biology*, vol. 18, no. 5, pp. 285–298, 2017.
- [7] F. Wippich, B. Bodenmiller, M. G. Trajkovska, S. Wanka, R. Aebersold, and L. Pelkmans, “Dual specificity kinase DYRK3 couples stress granule condensation/dissolution to mTORC1 signaling,” *Cell*, vol. 152, no. 4, pp. 791–805, 2013.
- [8] A. Patel, L. Malinowska, S. Saha et al., “ATP as a biological hydrotrope,” *Science*, vol. 356, no. 6339, pp. 753–756, 2017.
- [9] S. Jain, J. R. Wheeler, R. W. Walters, A. Agrawal, A. Barsic, and R. Parker, “ATPase-modulated stress granules contain a diverse proteome and substructure,” *Cell*, vol. 164, no. 3, pp. 487–498, 2016.
- [10] A. A. Hyman, C. A. Weber, and F. Julicher, “Liquid-liquid phase separation in biology,” *Annual Review of Cell and Developmental Biology*, vol. 30, no. 1, pp. 39–58, 2014.
- [11] C. P. Brangwynne, P. Tompa, and R. V. Pappu, “Polymer physics of intracellular phase transitions,” *Nature Physics*, vol. 11, no. 11, pp. 899–904, 2015.
- [12] P. J. Flory, “Thermodynamics of high polymer solutions,” *The Journal of Chemical Physics*, vol. 10, no. 1, pp. 51–61, 1942.
- [13] R. Nelson, M. R. Sawaya, M. Balbirnie et al., “Structure of the cross-beta spine of amyloid-like fibrils,” *Nature*, vol. 435, no. 7043, pp. 773–778, 2005.
- [14] J. Li, T. McQuade, A. B. Siemer et al., “The RIP1/RIP3 necrosome forms a functional amyloid signaling complex required for programmed necrosis,” *Cell*, vol. 150, no. 2, pp. 339–350, 2012.
- [15] R. Ciuffa, T. Lamark, A. K. Tarafder et al., “The selective autophagy receptor p 62 forms a flexible filamentous helical scaffold,” *Cell Reports*, vol. 11, no. 5, pp. 748–758, 2015.
- [16] K. A. Burke, A. M. Janke, C. L. Rhine, and N. L. Fawzi, “Residue-by-residue view of in vitro FUS granules that bind the C-terminal domain of RNA polymerase II,” *Molecular Cell*, vol. 60, no. 2, pp. 231–241, 2015.
- [17] S. Frey, R. P. Richter, and D. Gorlich, “FG-rich repeats of nuclear pore proteins form a three-dimensional meshwork with hydrogel-like properties,” *Science*, vol. 314, no. 5800, pp. 815–817, 2006.
- [18] Y. Lin, D. S. W. Protter, M. K. Rosen, and R. Parker, “Formation and maturation of phase-separated liquid droplets by RNA-binding proteins,” *Molecular Cell*, vol. 60, no. 2, pp. 208–219, 2015.

- [19] I. R. Mackenzie, A. M. Nicholson, M. Sarkar et al., “TIA1 mutations in amyotrophic lateral sclerosis and frontotemporal dementia promote phase separation and alter stress granule dynamics,” *Neuron*, vol. 95, no. 4, pp. 808–816.e9, 2017.
- [20] H. Wu and M. Fuxreiter, “The structure and dynamics of higher-order assemblies: amyloids, signalosomes, and granules,” *Cell*, vol. 165, no. 5, pp. 1055–1066, 2016.
- [21] T. J. Nott, E. Petsalaki, P. Farber et al., “Phase transition of a disordered nuage protein generates environmentally responsive membraneless organelles,” *Molecular Cell*, vol. 57, no. 5, pp. 936–947, 2015.
- [22] R. M. Vernon, P. A. Chong, B. Tsang et al., “Pi-Pi contacts are an overlooked protein feature relevant to phase separation,” *eLife*, vol. 7, 2018.
- [23] P. Li, S. Banjade, H. C. Cheng et al., “Phase transitions in the assembly of multivalent signalling proteins,” *Nature*, vol. 483, no. 7389, pp. 336–340, 2012.
- [24] T. S. Harmon, A. S. Holehouse, M. K. Rosen, and R. V. Pappu, “Intrinsically disordered linkers determine the interplay between phase separation and gelation in multivalent proteins,” *eLife*, vol. 6, 2017.
- [25] J. Fei, M. Jadalilha, T. S. Harmon et al., “Quantitative analysis of multilayer organization of proteins and RNA in nuclear speckles at super resolution,” *Journal of Cell Science*, vol. 130, no. 24, pp. 4180–4192, 2017.
- [26] P. Tompa and M. Fuxreiter, “Fuzzy complexes: polymorphism and structural disorder in protein-protein interactions,” *Trends in Biochemical Sciences*, vol. 33, no. 1, pp. 2–8, 2008.
- [27] M. Fuxreiter, I. Simon, and S. Bondos, “Dynamic protein-DNA recognition: beyond what can be seen,” *Trends in Biochemical Sciences*, vol. 36, no. 8, pp. 415–423, 2011.
- [28] M. Fuxreiter and P. Tompa, “Fuzzy complexes: a more stochastic view of protein function,” *Advances in Experimental Medicine and Biology*, vol. 725, pp. 1–14, 2012.
- [29] M. Fuxreiter, “Fuzziness: linking regulation to protein dynamics,” *Molecular BioSystems*, vol. 8, no. 1, pp. 168–177, 2012.
- [30] R. Sharma, Z. Raduly, M. Miskei, and M. Fuxreiter, “Fuzzy complexes: specific binding without complete folding,” *FEBS Letters*, vol. 589, no. 19, PartA, pp. 2533–2542, 2015.
- [31] M. Miskei, A. Gregus, R. Sharma, N. Duro, F. Zsolyomi, and M. Fuxreiter, “Fuzziness enables context dependence of protein interactions,” *FEBS Letters*, vol. 591, no. 17, pp. 2682–2695, 2017.
- [32] M. Fuxreiter, “Fuzziness in protein interactions—a historical perspective,” *Journal of Molecular Biology*, vol. 430, no. 16, pp. 2278–2287, 2018.
- [33] M. Miskei, C. Antal, and M. Fuxreiter, “FuzDB: database of fuzzy complexes, a tool to develop stochastic structure-function relationships for protein complexes and higher-order assemblies,” *Nucleic Acids Research*, vol. 45, D1, pp. D228–D235, 2017.
- [34] R. van der Lee, M. Buljan, B. Lang et al., “Classification of intrinsically disordered regions and proteins,” *Chemical Reviews*, vol. 114, no. 13, pp. 6589–6631, 2014.
- [35] M. Arbesu, G. Iruela, H. Fuentes, J. M. C. Teixeira, and M. Pons, “Intramolecular fuzzy interactions involving intrinsically disordered domains,” *Frontiers in Molecular Biosciences*, vol. 5, p. 39, 2018.
- [36] J. G. Olsen, K. Teilum, and B. B. Kragelund, “Behaviour of intrinsically disordered proteins in protein-protein complexes with an emphasis on fuzziness,” *Cellular and Molecular Life Sciences*, vol. 74, no. 17, pp. 3175–3183, 2017.
- [37] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [38] H. Singh, M. M. Gupta, T. Meitzler et al., “Real-life applications of fuzzy logic,” *Advances in Fuzzy Systems*, vol. 2013, Article ID 581879, 3 pages, 2013.
- [39] A. Borgia, M. B. Borgia, K. Bugge et al., “Extreme disorder in an ultrahigh-affinity protein complex,” *Nature*, vol. 555, no. 7694, pp. 61–66, 2018.
- [40] B. A. Shoemaker, J. J. Portman, and P. G. Wolynes, “Speeding molecular recognition by using the folding funnel: the fly-casting mechanism,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 16, pp. 8868–8873, 2000.
- [41] M. Dosnon, D. Bonetti, A. Morrone et al., “Demonstration of a folding after binding mechanism in the recognition between the measles virus N-TAIL and X domains,” *ACS Chemical Biology*, vol. 10, no. 3, pp. 795–802, 2015.
- [42] S. Boeynaems, S. Alberti, N. L. Fawzi et al., “Protein phase separation: a new phase in cell biology,” *Trends in Cell Biology*, vol. 28, no. 6, pp. 420–435, 2018.

Research Article

An Improved Differential Evolution Algorithm for a Multicommodity Location-Inventory Problem with False Failure Returns

Congdong Li ¹, Hao Guo ¹, Ying Zhang ², Shuai Deng ³, and Yu Wang⁴

¹School of Management, Jinan University, Guangzhou, Guangdong 510632, China

²Department of Industrial and Systems Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA

³School of Economic Management, Hunan Institute of Technology, Hengyang 421001, China

⁴International Business School, Jinan University, Zhuhai, Guangdong 519070, China

Correspondence should be addressed to Hao Guo; haoguo8701@gmail.com and Ying Zhang; yz29@buffalo.edu

Received 29 December 2017; Revised 26 September 2018; Accepted 15 October 2018; Published 24 October 2018

Academic Editor: Laszlo T. Koczy

Copyright © 2018 Congdong 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.

Customer returns are a common phenomenon in many industries, and they have a significant impact on business organizations and their supply chains. False failure returns are returned products that have no functional or cosmetic defects, and they represent a large body of customer returns in practice. In this paper, we develop a mixed-integer nonlinear programming model to study a multicommodity location-inventory problem in a forward-reverse logistics network. This model minimizes the total cost in this network by considering false failure returns, and it also considers many real-world business scenarios in forward and reverse logistics flows. Moreover, we design a new heuristic approach to solve the model efficiently. Finally, numerical experiments are conducted to validate our solution approach and provide meaningful managerial insights.

1. Introduction

Reverse logistics is an emerging topic that has attracted many research attentions because of the great importance of customer returns in today's business. Nowadays, the percentage of customer returns in total sales varies from 5% to 9% for durable goods, and it is up to 35% for fashion apparel products [1]. Since customer returns are more frequent in e-commerce, return rate in online shopping has exceeded 18% of total sales, and especially, it reaches 74% for fashion products [2]. In the retail industry, the value of customer products has exceeded \$100 billion per year in the United States [3]. Although many products are returned because of quality issues, a lot of items that have no defects can still be returned for various reasons such as installation difficulties, bad user experience, or price competition. Those items represent a large body of customer returns, and they are defined as "false failure returns" [4].

Forward-reverse logistics network consist of forward flows that are directed from suppliers to customers and

reverse flows of returned products from customers to suppliers. Since network design can contribute to reducing 60% the cost of a firm [5], Supply chain network design is an important strategic decision in supply chain management, and it is a big challenge to business managers. For example, many big companies such as Kodak, Xerox, and General Motors have developed their reverse logistics systems and those efforts are very successful [6]. Currently, many business decisions are made jointly to improve the efficiency of business organizations and their supply chains. Location-inventory problem (LIP) [7] is such a problem that integrates strategical facility location decisions and tactical inventory management decisions, and it can be studied by incorporating many real-world business scenarios to improve the performance of a logistics system.

In this paper, we study a multicommodity location-inventory problem (MCLIP) in a forward-reverse logistics network in which the reverse flow consists of false failure returns. This problem is formulated as a mixed-integer nonlinear programming model whose objective function

is to minimize the total cost in such a network, and it incorporates many business decisions and processes such as facility locations, return processing, and order fulfillment. More specifically, the solution to this problem will determine the number and locations of hybrid distribution-collection centers (HDCCs), how to use those HDCCs to fulfill orders and collect returns, and how to replenish inventories in those HDCCs. Due to the complexity of LIPs, we also design an improved differential evolution algorithm (IDE) to solve the MCLIP efficiently. Computational results are presented to validate the solution approach, and research implications and managerial insights are also discussed.

The rest of this paper is organized as follows: In Section 2, we review related research works in the literature. In Section 3, we describe the research problem and present the mathematical model. In Section 4, we propose a new heuristic approach to solve the problem efficiently. In Section 5, we show computational results and discuss managerial insights. Section 6 concludes the paper and provides directions for future research.

2. Literature Review

The design of logistics networks involves several core components such as facility location and inventory management. LIPs have been extensively studied because of their great importance in practice, and Farahani et al. [8] provide a comprehensive review of such works in the literature. In this section, we review the research works on LIPs from the perspective of forward and reverse logistics, respectively.

In the literature, most LIPs are studied under forward logistics networks. For example, Daskin et al. [7] study a LIP with the risk-pooling effect and design a Lagrangian relaxation algorithm as the solution approach. Shen et al. [9] use the same integer programming model as that in Daskin et al. [7] and convert it into a set-covering integer programming model and then solve it by column generation. Moreover, LIPs are studied by incorporating many business scenarios and processes. For example, Ozsen et al. [10] consider capacitated warehouses and the impact of multisourcing in addition to Shen et al. [9]. Amiri-Aref et al. [11] study multisourcing location-inventory problem with stochastic demand. Silva et al. [12] and Qu et al. [13] study the stochastic location-inventory problems under different replenishment policies. Mousavi et al. [14] study a seasonal multiple-product location allocation and inventory control problem in a planning horizon consisting of multiple periods. Since business environments are always dynamic, demand uncertainty [15–17] and different risks [18] are also considered.

It is emerging to study LIPs in the reverse logistics. Since reverse logistics is an important part of closed-loop supply chains (CLSCs), those works are usually conducted with CLSCs. For example, Diabat et al. [19] study a closed-loop location-inventory problem by considering spare parts, Asl-Najafi et al. [20] study a dynamic closed-loop location-inventory problem under disruption risk, and Kaya et al. [21] and Guo et al. [22] extend such studies by incorporating

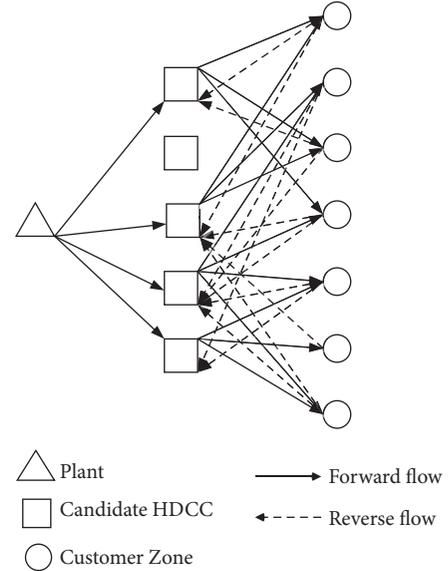


FIGURE 1: A forward-reverse logistics network with false failure returns.

pricing and vehicle routing decisions in CLSCs, respectively. Since returned products are usually sold to secondary markets in practice, LIPs are also studied by considering CLSCs and secondary markets [23, 24].

Although LIPs have been extensively studied from the perspective of reverse logistics, those works are still very limited because the types of customer returns are rarely considered. This paper studies a LIP in a forward-reverse logistics network in which the reverse logistics flow only consists of false failure returns. To the best of our knowledge, this is the first paper that considers false failure returns in the LIP literature, and it makes a significant contribution to the literature because of the great importance of false failure returns in practice.

3. Mathematical Model

3.1. Problem Description. We consider a three-echelon supply chain network in which a manufacturing plant ships multiple commodities to customer zones via a set of facilities. Moreover, returned products will be collected from customers and shipped to the same facilities. Therefore, those facilities are hybrid distribution-collection centers (HDCCs) [25, 26], and they have many advantages such as cost saving and pollution reduction by improving the efficiency of workflows. In this network, we assume that only false failure returns will be collected. Since those returns do not have any quality issues, they will not be sent to the plant for repair, and instead, they will be resold in the market after some simple processes (e.g., inspection, repackaging, etc.) in HDCCs. Figure 1 shows the forward-reverse logistics network under study. In this network, forward flows will be unidirectional from the plant to HDCCs and then from HDCCs to customer zones, but reverse flows will only be directed from customer zones to

HDCCs because false failure returns will not be sent to the plant for repair.

This paper studies a multicommodity location-inventory problem (MCLIP) in the network shown in Figure 1, and we make the following assumptions to simplify the problem: (i) customer zones are predetermined and fixed; (ii) the returned products will be the same as new items when they are resold to the market; (iii) the distances between HDCCs and customer zones are Euclidean distances. In practice, this problem can be solved to answer the following questions:

- (1) What are the best locations for HDCCs in this network?
- (2) How to assign those HDCCs to customer zones in the forward and reverse networks, respectively?
- (3) How to optimize inventory replenishment decisions at HDCCs?

3.2. Notations

Sets

- W : set of commodities, indexed by w ;
 R : set of candidate HDCC locations, indexed by r ;
 I : set of customer zones, indexed by i .

Parameters

- a_r : fixed (yearly) cost of building a HDCC at location r , for each $r \in R$;
 b_{rw} : fixed administrative and handling cost of placing an order for product w at HDCC r , for each $r \in R$ and $w \in W$;
 c_{rw} : basic transportation cost per shipment of product w from the plant to HDCC r , for each $r \in R$ and $w \in W$;
 e_{rw} : additional transportation cost per unit of product w from the plant to HDCC r , for each $r \in R$ and $w \in W$;
 d_{riw} : shipping cost per unit of product w from HDCC r to customer zone i , for each $r \in R$, $i \in I$ and $w \in W$;
 f_{rw} : repackaging/refurbishing cost per unit of returned product w at HDCC r , for each $r \in R$ and $w \in W$;
 g_{rw} : inspection cost per unit of returned product w at HDCC r , for each $r \in R$ and $w \in W$;
 h_{rw} : (yearly) holding cost per unit of product w at HDCC r , for each $r \in R$ and $w \in W$;
 q_w : return rate of product w , for each $w \in W$;
 μ_{iw} : mean (daily) demand of product w at customer zone i , for each $i \in I$ and $w \in W$;
 σ_{iw}^2 : variance of (daily) demand of product w at customer zone i , for each $i \in I$ and $w \in W$;
 α : desired percentage of market demand satisfied;
 z_α : standard normal deviation such that $P(z \leq z_\alpha) = \alpha$;
 L : order lead time in days at HDCCs;

λ : working days per year;

D_{rw} : mean (yearly) demand of product w at HDCC r , for each $r \in R$ and $w \in W$.

Decision Variables

Q_{rw} : order quantity of product w per order at HDCC r ;

$X_r = 1$ if opening one HDCC at location r , and 0 otherwise, for each $r \in R$;

$Y_{riw} = 1$ if HDCC r fulfills demand at customer zone i , and 0 otherwise, for each $r \in R$, $i \in I$ and $w \in W$;

$Z_{irw} = 1$ if HDCC r collects returns from customer zone i , and 0 otherwise, for each $r \in R$, $i \in I$ and $w \in W$.

3.3. The Model. In this paper, the research problem is formulated by a mixed-integer nonlinear programming model. The objective of this model is to minimize the total cost in the forward-reverse logistics network, and it comprises (i) location costs, (ii) shipping costs from HDCCs to customer zones, (iii) inventory costs including working inventory and safety stock costs, and (iv) return costs.

3.3.1. Location Cost. Obviously, the total location cost (C_L) can be written as

$$C_L = \sum_{r \in R} a_r X_r. \quad (1)$$

3.3.2. Shipping Cost from HDCCs to Customer Zones. The total shipping cost (C_T) from HDCCs to customer zones in the forward logistics flow can be expressed as follows:

$$C_T = \lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} \mu_{iw} d_{riw} Y_{riw}. \quad (2)$$

3.3.3. Inventory Cost. Inventory strategy plays an important role in supply chain management. In this study, we assume that HDCCs order new products from the plant by using an (R, Q) policy with type I service, and the total inventory cost includes working inventory and safety stock costs. The (R, Q) model can be approximated by an economic order quantity (EOQ) model [9], and Axsater [27] shows that the maximum relative error introduced by using the EOQ model instead of the (Q, r) policy is 0.118. Therefore, we use an EOQ model to optimize the inventory strategy in this study, and the order frequency and quantity at a HDCC will be calculated by using the total demand of the customer zones served by this HDCC.

In general, the working inventory cost includes (1) fixed cost of placing orders, (2) shipping cost from the plant to HDCCs, and (3) holding cost of the working inventory. The individual cost terms are formulated as follows.

(1) Order Cost. The annual fixed cost of placing orders for product w from HDCC r to the plant is given by $b_{rw}(D_{rw}/Q_{rw})$.

(2) *Shipping Cost from the Plant to HDCCs.* Since false failure returns will be resold to the market after they are processed in HDCCs, shipping quantities from the plant to HDCCs will equal the difference between market demands and return volumes. Therefore, the total shipping cost from the plant to HDCC r per year is given by

$$c_{rw} \frac{D_{rw}}{Q_{rw}} + \lambda e_{rw} \left(\sum_{i \in I} \mu_{iw} Y_{riw} - \sum_{i \in I} q_w \mu_{iw} Z_{irw} \right). \quad (3)$$

(3) *Holding Cost of the Working Inventory.* In this study, the working inventory comes from two sources which are the new products ordered from the plant and the returned products collected from customer zones. Therefore, the forward and reverse logistics flows will be integrated at HDCCs, and the average inventory of product w at HDCC r during period T can be expressed as follows:

$$\begin{aligned} \frac{1}{T} \int_0^T \left(S_{rw} - \frac{S_{rw} - q_w S_{rw}}{T} \right) dt &= S_{rw} - \frac{1}{2} \theta_{rw} T \\ &= \frac{Q_{rw}}{1 - q_w} - \frac{(1 - q_w)}{2} S_{rw} = \frac{(1 + q_w)}{2(1 - q_w)} Q_{rw}, \end{aligned} \quad (4)$$

where S_{rw} is the mean daily demand of product w at HDCC r . Consequently, the holding cost of the working inventory is given by

$$\sum_{i \in I} \frac{(1 + q_w) h_{rw}}{2(1 - q_w)} Q_{rw}. \quad (5)$$

Using Eppen's risk-pooling result [28], the amount of safety stock to ensure that stockouts occur with a probability of α or less is $z_\alpha \sqrt{L \sum_{i \in I} \sigma_{iw}^2 Y_{riw}}$, and the safety stock cost at HDCC r is given by $h_{rw} z_\alpha \sqrt{L \sum_{i \in I} \sigma_{iw}^2 Y_{riw}}$.

Therefore, the total annual inventory cost (C_I) of product w at HDCC r is given by

$$\begin{aligned} C_I &= b_{rw} \frac{D_{rw}}{Q_{rw}} + c_{rw} \frac{D_{rw}}{Q_{rw}} \\ &+ \lambda e_{rw} \left(\sum_{i \in I} \mu_{iw} Y_{riw} - \sum_{i \in I} q_w \mu_{iw} Z_{irw} \right) \\ &+ \sum_{i \in I} \frac{(1 + q_w) h_{rw}}{2(1 - q_w)} Q_{rw} + h_{rw} z_\alpha \sqrt{L \sum_{i \in I} \sigma_{iw}^2 Y_{riw}}. \end{aligned} \quad (6)$$

3.3.4. *Return Cost.* Returns costs are incurred by customer returns, and they include inspection cost, repackaging cost, and shipping cost from customer zones to HDCCs.

(1) *Inspection Cost.* Returned products will be inspected at HDCCs, and the total inspection cost is given by $\lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} q_w \mu_{iw} g_{rw} Z_{riw}$.

(2) *Repackaging Cost.* The returned products also need to be repackaged before they are resold to the market, and the total repackaging cost is given by $\lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} q_w \mu_{iw} f_{rw} Z_{riw}$.

(3) *Shipping Cost from Customer Zones to HDCCs.* The total shipping cost from customer zones to HDCCs is given by $\lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} q_w \mu_{iw} d_{riw} Z_{riw}$.

Therefore, the total annual return cost (C_R) is given by

$$C_R = \lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} q_w \mu_{iw} (g_{rw} + f_{rw} + d_{riw}) Z_{riw}. \quad (7)$$

3.3.5. *Optimization Model.* Given the individual cost functions above, the research problem under study can be formulated by a mixed-integer nonlinear program as follows:

$$\begin{aligned} \min Z &= C_L + C_T + C_I + C_R = \sum_{r \in R} a_r X_r \\ &+ \lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} \mu_{iw} d_{riw} Y_{riw} + \sum_{r \in R} \sum_{w \in W} \left\{ b_{rw} \frac{D_{rw}}{Q_{rw}} \right. \\ &+ c_{rw} \frac{D_{rw}}{Q_{rw}} + \lambda e_{rw} \left(\sum_{i \in I} \mu_{iw} Y_{riw} - \sum_{i \in I} q_w \mu_{iw} Z_{irw} \right) \\ &+ \sum_{i \in I} \frac{(1 + q_w) h_{rw}}{2(1 - q_w)} Q_{rw} + h_{rw} z_\alpha \sqrt{L \sum_{i \in I} \sigma_{iw}^2 Y_{riw}} \left. \right\} \\ &+ \lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} q_w \mu_{iw} (g_{rw} + f_{rw} + d_{riw}) Z_{riw} \end{aligned} \quad (8)$$

subject to

$$\sum_{r \in R} X_r \geq 1; \quad (9)$$

$$\sum_{r \in R} Y_{riw} = 1, \quad \forall i \in I, \forall w \in W; \quad (10)$$

$$\sum_{r \in R} Z_{irw} = 1, \quad \forall i \in I, \forall w \in W; \quad (11)$$

$$\sum_{r \in R} \sum_{w \in W} Y_{riw} = |W|, \quad \forall i \in I; \quad (12)$$

$$\sum_{r \in R} \sum_{w \in W} Z_{irw} = |W|, \quad \forall i \in I; \quad (13)$$

$$Y_{riw} \leq X_r, \quad \forall i \in I, \forall r \in R, \forall w \in W; \quad (14)$$

$$Z_{irw} \leq X_r, \quad \forall i \in I, \forall r \in R, \forall w \in W; \quad (15)$$

$$\sum_{w \in W} \sum_{i \in I} Y_{riw} \geq X_r, \quad \forall r \in R; \quad (16)$$

$$\sum_{i \in I} Z_{irw} \leq M \sum_{i \in I} Y_{riw}, \quad \forall r \in R, \forall w \in W; \quad (17)$$

$$X_r \in \{0, 1\}, \quad \forall r \in R; \quad (18)$$

$$Y_{riw} \in \{0, 1\}, \quad \forall i \in I, \forall r \in R, \forall w \in W; \quad (19)$$

$$Z_{irw} \in \{0, 1\}, \quad \forall i \in I, \forall r \in R, \forall w \in W; \quad (20)$$

In this model, objective function (8) means that the total cost will be minimized. Constraint (9) means that at least one

HDCC will be built. Constraint (10) means that the demand of a product in a customer zone will be fulfilled by only one HDCC. Constraint (11) means that the returns of a product from a customer zone will be collected by only one HDCC. Constraint (12) means that all types of products will be provided to a customer zone by HDCCs in the forward flow. Constraint (13) means that all types of returned products will be collected from a customer zone by HDCCs in the reverse flow. Constraints (14) and (15) mean that a customer zone can be assigned to a HDCC in the forward and reverse logistics flows, respectively, only if this HDCC is built. Constraint (16) means that a HDCC will serve at least one customer zone in the forward logistics network after it is built. Constraint (17) means that a HDCC will collect the returns of a product in the reverse flow only if it fulfills the demand of this product in the forward flow. Constraints (18)–(20) indicate that X_r , Y_{riw} , and Z_{irw} are binary variables.

Note that decision variable Q_{rw} only appears in the objective function. Since the objective function is convex in Q_{rw} , its optimal value can be obtained by taking the partial derivative of the objective function with respect to it, which gives

$$Q_{rw}^* = \sqrt{\sum_{i \in I} \frac{2(1-q_w)(b_{rw} + c_{rw})D_{rw}}{(1+q_w)h_{rw}}}. \quad (21)$$

Therefore, the objective function can be rewritten as follows:

$$\begin{aligned} \min Z = & \lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} \mu_{iw} d_{riw} Y_{riw} \\ & + \sum_{r \in R} \sum_{w \in W} \sqrt{\sum_{i \in I} \frac{2\lambda h_{rw} (1+q_w) (b_{rw} + c_{rw}) \mu_{iw} Y_{riw}}{(1-q_w)}} \\ & + \lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} e_{rw} (\mu_{iw} Y_{riw} - q_w \mu_{iw} Z_{irw}) \\ & + \sum_{r \in R} \sum_{w \in W} h_{rw} z_\alpha \sqrt{L \sum_{i \in I} \sigma_{iw}^2 Y_{riw}} + \sum_{r \in R} a_r X_r \\ & + \lambda \sum_{r \in R} \sum_{w \in W} \sum_{i \in I} q_w \mu_{iw} (g_{rw} + f_{rw} + d_{riw}) Z_{irw} \end{aligned} \quad (22)$$

4. Solution Approach

Location problems are NP-hard in general [29], and LIPs can be more complicated. Evolutionary algorithms are widely used to solve LIPs, and differential evolution (DE), which was first introduced by Storn and Price [30], is popular method to solve nonlinear and nonconvex functions [31]. Although DE has a strong global search ability to solve such problems, its performance is not always guaranteed due to its weaknesses such as the poor local search ability. To obtain a more stable and effective approach, we design an improved differential evolution algorithm (IDE) to solve MCLIP. More specifically, IDE enhances the performance of DE by two ways: First, IDE introduces opposition-based learning (OBL) [32] to generate better initial populations. Second, it uses a novel mutation

TABLE 1: IDE notations.

Notation	Explanation
Np	population size
N	dimension (i.e., number of customer zones in this study)
G	maximum number of generations for evolution
K	number of consecutive iterations if no better individual
$Z^{i,g}$	solution vector of individuals i in generation g ($i=1, 2, \dots, Np$)
$VZ^{i,g}$	mutant vector of individuals i in generation g ($i=1, 2, \dots, Np$)
$UZ^{i,g}$	trial vector of individuals i in generation g ($i=1, 2, \dots, Np$)
$F^{i,g}$	mutation factor of individuals i in generation g ($i=1, 2, \dots, Np$)
CR	crossover factor

factor F based on the Gaussian distribution $N(0,1)$ to enhance the diversity of populations.

4.1. Individuals and Operators. Generally, a DE algorithm has four main operations: initialization, mutation, crossover, and selection. After an initial population is created in the initialization step, the algorithm will execute mutation, crossover, and selection operations iteratively to improve the population until a stop criterion is satisfied. IDE follows the same procedure, and it uses notations shown in Table 1.

4.1.1. Initialization. When DE is applied to solve optimization problems, an encoding-decoding scheme will be needed to convert individuals in DE to the solutions to the optimization model, and vice versa. In IDE, an individual is represented by the matrix shown in (23).

$$Z^{i,g} = \begin{pmatrix} x_{1,1}^{i,g}, x_{1,2}^{i,g}, \dots, x_{1,N}^{i,g}, y_{1,N+1}^{i,g}, y_{1,N+2}^{i,g}, \dots, y_{1,2N}^{i,g} \\ \dots \\ x_{W,1}^{i,g}, x_{W,2}^{i,g}, \dots, x_{W,N}^{i,g}, y_{W,N+1}^{i,g}, y_{W,N+2}^{i,g}, \dots, y_{W,2N}^{i,g} \end{pmatrix}, \quad (23)$$

$i = 1, 2, \dots, Np; g = 1, 2, \dots, G;$

This matrix has W rows and $2N$ columns. For the research problem under study, W and N denote the numbers of commodities and customers, respectively, and the first and last N columns are related to the forward and reverse flows, respectively. More specifically, if $x_{w,j}^{i,g} = r$, then the demand of product w in customer zone j will be fulfilled by HDCC r in the forward flow, and if $y_{w,j}^{i,g} = r$, then the returns of product w from customer zone j will be collected by HDCC r in the reverse flow.

Figure 2 shows an example matrix for a MCLIP that has two commodities, ten customer zones, and five candidate HDCC locations. For product 1, HDCC 1 fulfills demands in customer zones 1, 6, and 10 and collects returns from

		Forward logistics										Reverse logistics									
Customer Product	Customer	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
	Product 1		1	3	3	2	3	1	2	3	5	1	1	2	5	3	3	2	1	3	1
Product 2		2	2	3	5	3	2	3	5	2	5	3	2	2	3	2	5	5	2	5	3

FIGURE 2: An example individual.

customer zones 1, 7, and 9; HDCC 2 fulfills demands in customer zones 4 and 7 and collects returns from customer zones 2 and 6; HDCC 3 fulfills demands in customer zones 2, 3, 5, and 8 and collects returns from customer zones 4, 5, and 8; HDCC 5 fulfills the demand in customer zone 9 and collects returns from customer zones 3 and 10. For product 2, location decisions can be also explained in the same manner.

The aim of encoding is transforming the initial individual to practical solution of LIP. The formulation of decoding is as follows.

To decode an individual to a solution to the optimization problem, we can use a mechanism shown in (24) and (25).

$$x_{w,j}^{i,0} = \text{round}(x^L + \text{rand} \cdot (x^U - x^L)); \quad (24)$$

$i = 1, 2, \dots, Np; j = 1, 2, \dots, N; w = 1, 2, \dots, W;$

$$y_{w,j}^{i,0} = U(\text{round}(y^L + \text{rand} \cdot (\text{length}(U) - 1))); \quad (25)$$

$i = 1, 2, \dots, Np; j = N + 1, N + 2, \dots, 2N; w = 1, 2, \dots, W;$

where $U = \text{unique}(x_{w,1}^{i,g}, x_{w,2}^{i,g}, \dots, x_{w,N}^{i,g})$ are distinct HDCCs that provide product w in the forward flow, rand is a random variable that is uniformly distributed on $[0, 1]$, x^L and x^U

are the lower and upper bounds of $x_{w,j}^{i,0}$, and round is the rounding function. Obviously, we have $x^L = y^L = 1$, and x^U is decided by the number of candidate HDCC locations. The expression of $y_{w,j}^{i,0}$ ensures that product w can be returned to a HDCC in the reverse flow only if it provides this product in the forward flow, which is reinforced by Constraint (17) in the optimization model shown in Section 3.3.5.

IDE introduce a new mechanism based on OBL to generate better initial populations. More specifically, IDE will improve an initial population that is generated randomly by checking its opposite population by the following steps.

Step 1. Generate an initial population $Z^{i,0}$ randomly:

$$Z^{i,0} = \begin{pmatrix} x_{1,1}^{i,0}, x_{1,2}^{i,0}, \dots, x_{1,N}^{i,0}, y_{1,N+1}^{i,0}, y_{1,N+2}^{i,0}, \dots, y_{1,2N}^{i,0} \\ \dots \\ x_{W,1}^{i,0}, x_{W,2}^{i,0}, \dots, x_{W,N}^{i,0}, y_{W,N+1}^{i,0}, y_{W,N+2}^{i,0}, \dots, y_{W,2N}^{i,0} \end{pmatrix}, \quad (26)$$

$i = 1, 2, \dots, Np;$

Step 2. Obtain its opposite population $\overline{Z}^{i,0}$:

$$\overline{Z}^{i,0} = \begin{pmatrix} \overline{x_{1,1}^{i,0}}, \overline{x_{1,2}^{i,0}}, \dots, \overline{x_{1,N}^{i,0}}, \overline{y_{1,N+1}^{i,0}}, \overline{y_{1,N+2}^{i,0}}, \dots, \overline{y_{1,2N}^{i,0}} \\ \dots \\ \overline{x_{W,1}^{i,0}}, \overline{x_{W,2}^{i,0}}, \dots, \overline{x_{W,N}^{i,0}}, \overline{y_{W,N+1}^{i,0}}, \overline{y_{W,N+2}^{i,0}}, \dots, \overline{y_{W,2N}^{i,0}} \end{pmatrix} \quad (27)$$

$$\overline{x_{w,j}^{i,0}} = x^L + x^U - x_{w,j}^{i,0}; \quad i = 1, 2, \dots, Np; j = 1, 2, \dots, N; w = 1, 2, \dots, W;$$

where $\overline{x_{w,j}^{i,0}}$ is the entry in row w and column j of the i^{th} individual in the opposite population and $\overline{y_{w,j}^{i,0}}$ ($i=1, 2, \dots, Np; j=N+1, N+2, \dots, 2N; w=1, 2, \dots, W$) is calculated by (25).

Step 3. Select the Np best solutions from $\{Z^{i,0} \cup \overline{Z}^{i,0}\}$ as initial population.

4.1.2. Mutation. The mutation strategy plays an important role DE to enhance the global optimization ability and convergence speed. Usually, DE will use DE/rand/1/bin as the mutation operation that will produce a new vector by adding the weighted difference of two randomly selected vectors

to the third vector in generation g [30]. In this study, we design a new mutation strategy to improve the diversity of populations, and new mutant vectors $v x_{w,j}^{i,g+1}$ and $v y_{w,j}^{i,g+1}$ are generated by (28) and (29).

$$v x_{w,j}^{i,g+1} = \text{round}(x_{w,j}^{r1,g} + F^{i,g} \cdot (x_{w,j}^{r2,g} - x_{w,j}^{r3,g})); \quad (28)$$

$i = 1, 2, \dots, Np; j = 1, 2, \dots, N; w = 1, 2, \dots, W;$

$$v y_{w,j}^{i,g+1} = \text{round}(y_{w,j}^{r1,g} + F^{i,g} \cdot (y_{w,j}^{r2,g} - y_{w,j}^{r3,g})); \quad (29)$$

$i = 1, 2, \dots, Np; j = N + 1, N + 2, \dots, 2N; w = 1, 2, \dots, W;$

where $i \neq r1 \neq r2 \neq r3$, $r1, r2, r3$ are randomly selected from $\{1, 2, \dots, Np\}$, $F^{i,g} = F \cdot \xi^{i,g}$ and it amplifies the difference

between two vectors, F is an initial mutation factor, and $\xi^{i,g}$ is independently generated from the normal distribution $N(0, 1)$. In the mutation step, the mutation factor will be updated first and then used to generate new mutant vectors, which will improve local and global search abilities and generate better mutant vectors. If $v x_{w,j}^{i,g+1}$ and $v y_{w,j}^{i,g+1}$ in a new mutant vector exceed the upper and lower bounds, then the vector will be generated again within its range.

4.1.3. Crossover. The crossover operation mixes mutate and target vectors to generate trial vectors and increase the diversity of populations. In IDE, the trial vector is generated by (30) and (31).

$$u x_{w,j}^{i,g+1} = \begin{cases} v x_{w,j}^{i,g} & \text{rand}(i) < CR \\ x_{w,j}^{i,g} & \text{otherwise} \end{cases} \quad (30)$$

$$i = 1, 2, \dots, Np; \quad j = 1, 2, \dots, N; \quad w = 1, 2, \dots, W;$$

$$u y_{w,j}^{i,g+1} = \begin{cases} v y_{w,j}^{i,g} & \text{rand}(i) < CR \\ y_{w,j}^{i,g} & \text{otherwise} \end{cases} \quad (31)$$

$$i = 1, 2, \dots, Np; \quad j = N + 1, N + 2, \dots, 2N; \quad w = 1, 2, \dots, W;$$

where $\text{rand}(i)$ is random variable that is uniformly distributed on $[0, 1]$, and $CR \in [0, 1]$ is a predefined crossover rate.

4.1.4. Feasibility Correction. A new individual that is generated by mutation or crossover operations may not be feasible by violating Constraint (17). Therefore, a feasibility correction procedure is developed to exclude infeasible individuals. In this study, a new individual will be checked after it is generated. If Constraint (17) is violated, then it will be replaced by a new individual that is randomly generated by (25).

4.1.5. Selection. After a new population is generated, the selection operation will be executed to evaluate the objective values of all trial vectors. In IDE, a trial vector $f(UX^{i,g+1})$ will be compared with the corresponding target vector $f(X^{i,g})$ on their objective values by using a greedy criterion[30]. More specifically, the selection operation can be expressed as follows:

$$Z^{i,g+1} = \begin{cases} UZ^{i,g+1} & f(UZ^{i,g+1}) < f(Z^{i,g}) \\ Z^{i,g} & \text{otherwise.} \end{cases} \quad (32)$$

4.1.6. Stop Criterion. In DE, the mutation, crossover, and selection operations will be executed iteratively until a stop criterion is satisfied. In this study, IDE will stop until any of the criteria below is satisfied.

- (1) No better solution is found in consecutive K iterations.
- (2) The maximum number of iterations, i.e., G , is reached.

4.2. Algorithm Flow. In summary, IDE consists of the following steps.

Step 1. Set $g = 0$ and $k = 0$.

Step 2. Create an initial population that consists of Np individuals by applying OBL, and calculate the objective values of all individuals in the population.

Step 3. Execute mutation operation.

Step 4. Execute crossover operation.

Step 5. Perform feasibility correction.

Step 6. Execute selection operation.

Step 7. Set $g = g + 1$. Calculate the objective values of all individuals in the new population. If the minimal objective value is less than that in the old population, set $k = 1$. Otherwise, set $k = k + 1$.

Step 8. If $k = K$ or $g = G$, stop the algorithm. Otherwise, go to Step 4.

5. Computational Results

In this section, numerical analysis is presented to validate IDE and provide managerial insights. First, we show how to obtain the optimal setting in IDE for the best possible performance. Second, the solution to an example MCLIP is presented. Third, sensitivity analysis is conducted to study the influences of the parameters in MCLIP. Last, IDE is compared with Lingo 11 and DE to validate its performance. All the experiments are implemented by Java JDK 1.7 on a Windows PC (AMD A10-9600P RADEON R5, 10 COMPUTE CORES 4C+6G 2.40GHz; RAM: 4.00 GB DDR; OS: Windows 10).

To get started, we use an example MCLIP that consists of eight candidate HDCC locations, eighty customer zones, and three commodities. The candidate HDCC locations and customer zones are uniformly distributed on a grid of $[0,50] \times [0,50]$, and the parameters are shown in Table 2.

5.1. Parameter Analysis. The performance of DE can be significantly affected by its parameters or the combinations of those parameters. In this subsection, we test the performance of IDE under different F and CR values. In this experiment, we set $Np = 4S$ ($S = N * W$) and the algorithm was executed 30 times under each setting, and the numerical results are shown in Table 3.

From Table 3, we can see that F and CR have a significant impact on the searching ability and convergence speed of IDE, which are indicated by OS ratio and CPU time, respectively. Obviously, IDE has the best performance when $F = 0.1$ and $CR = 0.1$, and this setting will be used in the subsequent experiments.

5.2. Illustrative Example. This subsection shows the optimal to the example MCLIP. Table 4 shows the optimal solution which indicates HDCC locations and the corresponding customer zones in the forward and reverse flows, and Table 5

TABLE 2: MCLIP parameters.

Parameters	Value	Parameters	Value	Parameters	Value
a_r	$U[1000, 1500]$	b_{rw}	$U[5, 10]$	c_{rw}	10
e_{rw}	5	f_{rw}	$U[1, 3]$	g_{rw}	$U[1, 3]$
h_{rw}	$U[1, 3]$	q_w	$U[0.3, 0.6]$	μ_{iw}	$U[20, 30]$
σ_{iw}^2	$\sigma_i^2 = \mu_i$	z_α	1.96	L	1

Remark: $U[a, b]$ denotes the uniform distribution over $[a, b]$.

TABLE 3: IDE parameter test results.

		CR									
		F	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
OS Ratio	0.1	96.67%	56.67%	36.67%	3.33%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		28.42	22.74	20.13	18.90	17.79	17.16	17.15	16.78	17.33	
OS Ratio	0.2	96.67%	83.33%	23.33%	10.00%	3.33%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		42.01	35.09	31.17	29.26	27.49	26.76	26.41	26.22	27.18	
OS Ratio	0.3	0.00%	56.67%	50.00%	20.00%	3.33%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		47.40	46.46	43.12	40.61	38.78	37.51	36.65	36.72	37.19	
OS Ratio	0.4	0.00%	0.00%	0.00%	3.33%	6.67%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		46.89	47.66	46.78	46.04	45.43	44.88	44.63	44.37	44.32	
OS Ratio	0.5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		47.01	47.00	45.49	46.61	46.18	46.28	46.08	44.13	45.53	
OS Ratio	0.6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		45.89	47.56	47.17	46.26	44.70	46.78	47.01	45.96	44.89	
OS Ratio	0.7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		47.45	47.03	46.94	47.53	44.71	45.39	46.70	45.14	45.11	
OS Ratio	0.8	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		47.23	47.82	46.28	45.75	45.43	45.27	43.85	46.25	46.59	
OS Ratio	0.9	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
CPU Time		48.08	44.82	44.95	46.98	46.45	47.66	45.06	45.77	43.33	

Remark: in Table 3, the optimal solutions are obtained by Lingo 11, and ‘‘OS Ratio’’ means the percentage of finding optimal solutions by IDE.

shows the optimal value and individual costs in the optimal solution.

5.3. Sensitivity Analysis. The optimal solution of a MCLIP can be affected significantly by important business parameters. In this subsection, sensitivity analysis is conducted on q_w , d_{riw} , b_{rw} , c_{rw} , e_{rw} , h_{rw} , g_{rw} , and f_{rw} to study their influences. More specifically, a parameter will be tested by changing its value by (-30%, 30%) and fixing the other parameters and the numerical results are shown in Table 6.

From Table 6, we can see that optimal values can change significantly due to the change of those parameters. For example, the optimal value will decrease by 6.63% if q_w decreases by 30%, and the optimal value will increase by 6.65% if q_w increases by 30%. The influences of the parameters on individual costs are summarized as follows: When q_w increases, C_R will increase, C_I and will decrease, but C_L and C_T will not change. When d_{riw} increases, C_T and C_R will increase but C_L and C_I will not change. When b_{rw} , c_{rw} , e_{rw} , h_{rw} increase, only C_I will increase and the other costs will not change. When g_{rw} or f_{rw} increases, only C_I will increase. We can see that, in general, d_{riw} and q_w

have the most significant influences. Moreover, the optimal value is more sensitive when d_{riw} , q_w , and e_{rw} are relatively small. In practice, business managers may focus more on those parameters to stabilize the supply chain performance.

5.4. Performance Analysis. In this subsection, IDE is compared with Lingo 11 and DE to validate its performance in terms of solution accuracy and CPU time efficiency, and the three approaches are tested on a set of small- (i.e., $20 \times 5 \times 2/3/4$, $40 \times 6 \times 2/3/4$), medium- (i.e., $50 \times 5/6/7 \times 3$, $70 \times 6/7/8 \times 3$), and large-sized problems (i.e., $80 \times 8 \times 2/3/4$, $100 \times 8/9/10 \times 3$). Since population size Np is an important parameter in IDE and it may vary according to the problem sizes, its value needs to be set properly to get the best possible performance. According to Figures 3–5 (in which ‘‘red line’’ represents solution accuracy in terms of the percentage of finding the optimal solutions, and ‘‘blue line’’ represents the mean CPU time), IDE will have the best performance when $Np = 3S$, $4S$, $5S$ for small-, medium-, and large-sized problems, respectively. Therefore, we will use these settings in the subsequent experiments.

TABLE 4: Optimal solution to the example MCLIP.

HDCC Location	Commodity	Customer Zone (Forward Flow)	Customer Zone (Reverse Flow)
1	1	48,64,75	48,64,75
	2	48,64,75	48,64,75
	3	48,64,75	48,64,75
2	1	4,9,10,11,26,29,30,39,44,45,47,57,68,72	4,9,10,11,26,29,30,39,44,45,47,57,68,72
	2	4,9,10,11,26,29,30,39,44,45,47,57,68,72	4,9,10,11,26,29,30,39,44,45,47,57,68,72
	3	4,9,10,11,26,29,30,39,44,45,47,57,68,72	4,9,10,11,12,26,29,30,39,44,45,47,57,68,71,72
3	1	12,14,18,22,32,34,43,60,70,71	12,14,18,22,32,34,43,60,70,71
	2	12,14,18,22,32,34,43,60,70,71	12,14,18,22,32,34,38,43,60,70,71,79
	3	12,14,18,22,32,34,43,60,70,71	14,18,22,32,34,43,60,70
4	1	2,7,13,35,52,56,58,63,76,80	2,7,35,52,56,58,80
	2	2,7,13,35,52,56,58,63,76,80	2,7,13,35,52,56,58,63,76,80
	3	2,7,13,35,52,56,58,63,76,80	2,7,8,13,21,35,51,52,56,58,61,63,76,80
5	1	5,15,16,19,31,37,38,42,49,62,77,79	5,15,16,19,31,37,38,42,49,62,77,79
	2	5,15,16,19,31,37,38,42,49,62,77,79	5,15,16,19,31,37,42,49,62,77,
	3	5,15,16,19,31,37,38,42,49,62,77,79	5,15,16,19,31,37,38,42,49,62,77,79
6	1	3,6,17,20,23,24,36,67,69	3,17,20,23,24,36,67,69
	2	3,6,17,20,23,24,36,67,69	3,6,17,20,23,24,36,67,69
	3	3,6,17,20,23,24,36,67,69	3,6,17,20,23,24,36,67,69
7	1	8,21,25,28,46,51,53,54,61,65,73,74	6,8,13,21,25,27,28,46,51,53,54,61,63,65,73,74,76
	2	8,21,25,28,46,51,53,54,61,65,73,74	8,21,25,28,46,51,53,54,61,65,73,74
	3	8,21,25,28,46,51,53,54,61,65,73,74	25,28,46,53,54,65,73,74
8	1	1,27,33,40,41,50,55,59,66,78	1,33,40,41,50,55,59,66,78
	2	1,27,33,40,41,50,55,59,66,78	1,27,33,40,41,50,55,59,66,78
	3	1,27,33,40,41,50,55,59,66,78	1,27,33,40,50,55,59,66,78

TABLE 5: Optimal value and individual costs.

C_L	C_T	C_I	C_R	Total
10262	16651085.12	4665316.68	11617106.42	32943770.22

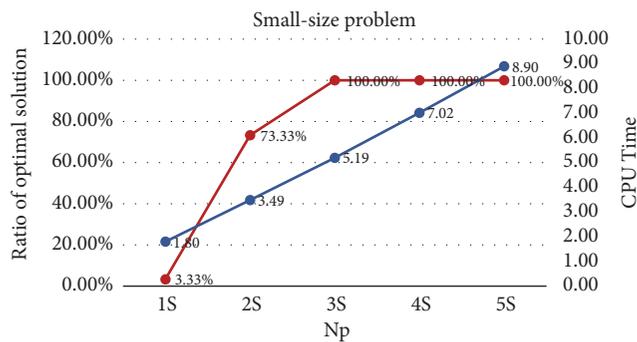


FIGURE 3: Analysis of Np for small-sized problems ($40 \times 6 \times 4$).

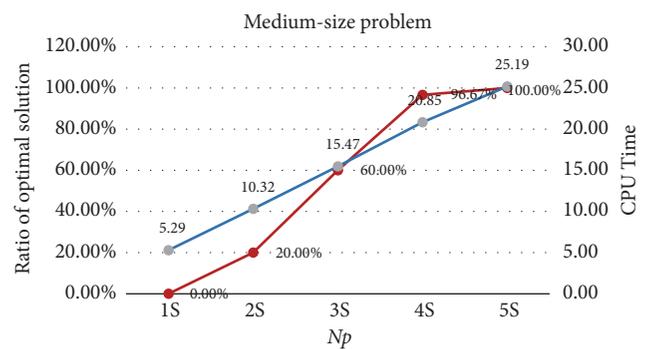


FIGURE 4: Analysis of Np for medium-sized problems ($70 \times 8 \times 3$).

Table 7 shows the comparison results between IDE, DE, and Lingo 11. To avoid biased results, IDE and DE were executed 30 times for each test instance. Therefore, in this table, “O.V.” means the optimal value, “O.V. (Best)” and “O.V. (Mean)” represent the best and mean optimal values in 30 runs, respectively, “OS Ratio” means the percentage of finding the optimal solutions, and “S.D.” means the standard deviation of optimal values in 30 runs.

From Table 7, we can see that IDE is the most effective and efficient method to solve MCLIPs, and its advantages over Lingo 11 and DE are summarized as follows:

- (1) The optimal values obtained by IDE and Lingo 11 were almost identical, but IDE has better runtimes than Lingo 11. This indicates that IDE is much more efficient than Lingo 11 with the same solution accuracy.

TABLE 6: Sensitivity analysis results.

Δ		-30.00%	-20.00%	-10.00%	0.00%	10.00%	20.00%	30.00%
q_w	C_I	5967659.67	5532973.87	5098809.85	4665316.678	4232715.54	3801356.65	3371846.74
	Gap	27.92%	18.60%	9.29%	0.00%	-9.27%	-18.52%	-27.73%
	C_R	8131974.50	9293685.14	10455395.78	11617106.42	12778817.06	13940527.71	15102238.35
	Gap	-30.00%	-20.00%	-10.00%	0.00%	10.00%	20.00%	30.00%
	OV	30760981.29	31488006.13	32215552.76	32943770.22	33672879.73	34403231.48	35135432.21
	Gap	-6.63%	-4.42%	-2.21%	0.00%	2.21%	4.43%	6.65%
d_{riw}	C_T	11655759.59	13320868.10	14985976.61	16651085.12	18316193.63	19981302.15	21646410.66
	Gap	-30.00%	-20.00%	-10.00%	0.00%	10.00%	20.00%	30.00%
	C_R	9141586.34	9969691.59	10793951.68	11617106.42	12437398.32	13255554.35	14073565.94
	Gap	-21.31%	-14.18%	-7.09%	0.00%	7.06%	14.10%	21.15%
	OV	25472924.60	27966138.37	30455506.97	32943770.22	35429170.64	37912435.18	40395555.28
	Gap	-22.68%	-15.11%	-7.55%	0.00%	7.54%	15.08%	22.62%
b_{rw}	C_I	4659545.89	4661513.59	4663436.18	4665316.678	4667157.74	4668961.77	4670730.94
	Gap	-0.12%	-0.08%	-0.04%	0.00%	0.04%	0.08%	0.12%
	OV	32937999.43	32939967.13	32941889.73	32943770.22	32945611.28	32947415.32	32949184.48
	Gap	-0.02%	-0.01%	-0.01%	0.00%	0.01%	0.01%	0.02%
c_{rw}	C_I	4657224.85	4660009.99	4662704.48	4665316.678	4667853.73	4670321.79	4672726.23
	Gap	-0.17%	-0.11%	-0.06%	0.00%	0.05%	0.11%	0.16%
	OV	32935678.39	32938463.54	32941158.03	32943770.22	32946307.27	32948775.33	32951179.77
	Gap	-0.02%	-0.02%	-0.01%	0.00%	0.01%	0.02%	0.02%
e_{rw}	C_I	3292753.68	3750274.68	4207795.68	4665316.678	5122837.68	5580358.68	6037879.68
	Gap	-29.42%	-19.61%	-9.81%	0.00%	9.81%	19.61%	29.42%
	OV	31571207.22	32028728.22	32486249.22	32943770.22	33401291.22	33858812.22	34316333.22
	Gap	-4.17%	-2.78%	-1.39%	0.00%	1.39%	2.78%	4.17%
h_{rw}	C_I	4650403.35	4655668.91	4660623.12	4665316.678	4669787.84	4674066.35	4678175.84
	Gap	-0.32%	-0.21%	-0.10%	0.00%	0.10%	0.19%	0.28%
	OV	32928856.89	32934122.46	32939076.67	32943770.22	32948241.39	32952519.89	32956629.38
	Gap	-0.05%	-0.03%	-0.01%	0.00%	0.01%	0.03%	0.04%
g_{rw}	C_R	11058612.56	11245420.25	11431721.41	11617106.42	11802239.42	11987372.42	12172505.42
	Gap	-4.81%	-3.20%	-1.60%	0.00%	1.59%	3.19%	4.78%
	OV	32385276.36	32572084.05	32758385.21	32943770.22	33128903.22	33314036.22	33499169.22
	Gap	-1.70%	-1.13%	-0.56%	0.00%	0.56%	1.12%	1.69%
f_{rw}	C_R	11146187.90	11303872.70	11461479.65	11617106.42	11770266.78	11922189.48	12073560.78
	Gap	-4.05%	-2.70%	-1.34%	0.00%	1.32%	2.63%	3.93%
	OV	32472851.69	32630536.49	32788143.45	32943770.22	33096930.58	33248853.28	33400224.58
	Gap	-1.43%	-0.95%	-0.47%	0.00%	0.46%	0.93%	1.39%

Remark: in Table 6, "OV" means the optimal value, and "Gap" means the change of the optimal value. Note that the gap will be zero if no parameter is changed.

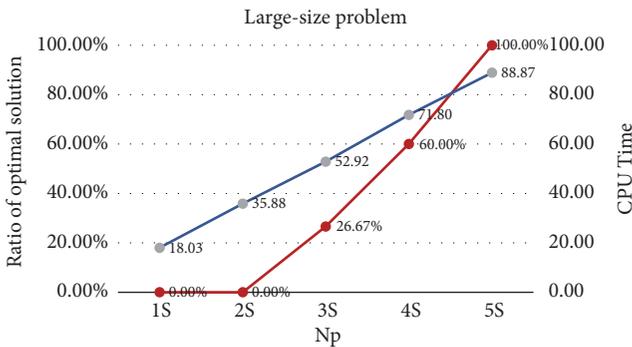


FIGURE 5: Analysis of N_p for large-sized problems ($100 \times 10 \times 3$).

(2) IDE and DE can achieve similar solution accuracies on small- and medium-sized problems. However, the optimal values from DE are significantly larger than those from IDE on large-sized problems. Particularly, DE cannot find optimal solutions for the problems whose sizes are over $(100 \times 9 \times 3)$. This indicates that IDE has a great global search ability and is more stable and consistent than DE.

(3) IDE has better runtimes than DE in most test instances, and it has a greater advantage on large-sized problems. This indicates that IDE is more efficient than DE.

TABLE 7: Lingo II versus DE versus IDE.

Instance	Lingo II			DE			IDE									
	I	R	W	O.V.	CPU Time (S)	O.V. (Best)	O.V. (Mean)	OS Ratio	CPU Time (S) - T1	S.D.	O.V. (Best)	O.V. (Mean)	OS Ratio	CPU Time (S) - T2	S.D.	(T1-T2) / T1
20	2	6987580	6	6987580.30	6997722.08	21/30	0.14	0.01	6987580.30	6987580.30	30/30	0.14	0.01	0.14	0.01	-3.95%
	3	10355510	15	10355509.01	10358913.98	26/30	0.34	0.05	10355509.01	10355509.01	30/30	0.35	0.03	0.35	0.03	-2.63%
	4	13487440	27	13487441.55	13488007.01	29/30	0.65	0.05	13487441.55	13487441.55	30/30	0.67	0.05	0.67	0.05	-3.08%
	2	12598710	53	12598709.04	12608013.39	16/30	1.01	0.06	12598709.04	12598750.74	29/30	0.89	0.04	0.89	0.04	11.68%
40	3	18642000	135	18642000.91	18643688.34	26/30	2.71	0.08	18642000.91	18642000.91	30/30	2.50	0.08	2.50	0.08	7.71%
	4	24367940	263	24367942.66	24368298.67	26/30	5.71	0.14	24367942.66	24367942.66	30/30	5.19	0.10	5.19	0.10	9.02%
	5	23313920	123	23313920.93	23316057.84	26/30	4.85	0.32	23313920.93	23313920.93	30/30	5.03	0.21	5.03	0.21	-3.68%
	6	23017930	229	23017934.37	23018947.56	27/30	6.78	0.16	23017934.37	23017934.37	30/30	6.05	0.12	6.05	0.12	10.76%
70	7	19471180	326	19471179.29	19471323.45	28/30	9.57	0.17	19471179.29	19471179.29	30/30	7.45	0.16	7.45	0.16	22.14%
	6	31652380	472	31652384.45	31652401.08	29/30	16.17	0.50	31652384.45	31652384.45	30/30	16.26	0.60	16.26	0.60	-0.56%
	7	30150360	1394	30150355.84	30150535.41	27/30	21.51	0.47	30150355.84	30150355.84	30/30	18.28	0.42	18.28	0.42	14.99%
	8	28295930	1993	28295927.33	28296407.75	25/30	22.89	0.43	28295927.33	28295939.60	29/30	20.85	0.53	20.85	0.53	8.92%
80	2	21939770	721	21939767.66	21939991.65	28/30	13.28	0.24	21939767.66	21939767.66	30/30	12.05	0.31	12.05	0.31	9.31%
	3	32943770	2974	32943770.22	32943973.24	29/30	41.51	1.15	32943770.22	32943770.22	30/30	35.76	0.72	35.76	0.72	13.83%
	4	43234620	>3600	43234618.55	43234779.15	29/30	91.60	1.84	43234618.55	43234618.55	30/30	74.05	1.35	74.05	1.35	19.15%
	8	39081930	>3600	39081579.93	39081760.66	28/30	115.95	2.07	39081579.93	39081579.93	30/30	69.93	1.45	69.93	1.45	39.69%
100	9	38614670	>3600	39767341.71	40015199.70	0/30	102.18	0.90	38614396.07	38614592.75	29/30	83.84	1.72	83.84	1.72	17.95%
	10	36450520	>3600	37621661.96	37827792.61	0/30	124.76	0.74	36450173.57	36450173.57	30/30	88.87	2.74	88.87	2.74	28.76%

According to the comparison results, we can see that IDE is the best approach to solve MCLIPs compared with DE and Lingo II.

6. Conclusion and Future Research

In this paper, we study a multicommodity location-inventory problem by considering false failure returns in a forward-reverse logistics network. To the best of our knowledge, this is the first work that considers false failure returns to optimize joint location-inventory decisions, and it contributes to the literature because of the great importance of false failure returns in practice.

This research can be extended in several directions: First, other types of returns such as defective items can be considered to make this study more practical. Second, the (R, Q) model is adopted to replenish inventories in this work. However, there are other popular inventory control policies in practice, and it will be interesting to explore them and compare their performances in forward-reverse logistics networks. Third, it will be valuable to study location-inventory-routing problems by considering false failure returns.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grants nos. 71672074 and 71772075.

References

- [1] V. D. R. Guide Jr., G. C. Souza, L. N. van Wassenhove, and J. D. Blackburn, "Time value of commercial product returns," *Management Science*, vol. 52, no. 8, pp. 1200–1214, 2006.
- [2] Z. Hua, H. Hou, and Y. Bian, "Optimal Shipping Strategy and Return Service Charge under No-Reason Return Policy in Online Retailing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3189–3206, 2017.
- [3] J. Stock, T. Speh, and H. Shear, "Many happy (product) returns," *Harvard Business Review*, vol. 80, no. 7, p. 16, 2002.
- [4] M. Ferguson, V. D. R. Guide Jr., and G. C. Souza, "Supply chain coordination for false failure returns," *Manufacturing and Service Operations Management*, vol. 8, no. 4, pp. 376–393, 2006.
- [5] T. P. Harrison, *The Practice of Supply Chain Management: Where Theory And Application Converge*, Springer Science & Business Media, 2005.
- [6] M. S. Pishvaei, F. Jolai, and J. Razmi, "A stochastic optimization model for integrated forward/reverse logistics network design," *Journal of Manufacturing Systems*, vol. 28, no. 4, pp. 107–114, 2009.
- [7] M. S. Daskin, C. R. Coullard, and Z. J. M. Shen, "An inventory-location model: formulation, solution algorithm and computational results," *Annals of Operations Research*, vol. 110, pp. 83–106, 2002.
- [8] R. Z. Farahani, H. Rashidi Bajgan, B. Fahimnia, and M. Kaviani, "Location-inventory problem in supply chains: A modelling review," *International Journal of Production Research*, vol. 53, no. 12, pp. 3769–3788, 2015.
- [9] Z.-J. M. Shen, C. R. Coullard, and M. S. Daskin, "A joint location-inventory model," *Transportation Science*, vol. 37, no. 1, pp. 40–55, 2003.
- [10] L. Ozsen, M. S. Daskin, and C. R. Coullard, "Facility location modeling and inventory management with multisourcing," *Transportation Science*, vol. 43, no. 4, pp. 455–472, 2009.
- [11] M. Amiri-Aref, W. Klibi, and M. Z. Babai, "The multi-sourcing location inventory problem with stochastic demand," *European Journal of Operational Research*, vol. 266, no. 1, pp. 72–87, 2018.
- [12] F. Silva and L. Gao, "A Joint Replenishment Inventory-Location Model," *Networks and Spatial Economics*, vol. 13, no. 1, pp. 107–122, 2013.
- [13] Q. Hui, W. Lin, and L. Rui, "A contrastive study of the stochastic location-inventory problem with joint replenishment and independent replenishment," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2061–2072, 2015.
- [14] S. M. Mousavi, N. Alikar, S. T. A. Niaki, and A. Bahreinnejad, "Optimizing a location allocation-inventory problem in a two-echelon supply chain network: a modified fruit fly optimization algorithm," *Computers & Industrial Engineering*, vol. 87, pp. 543–560, 2015.
- [15] M. a. Schuster Puga and J.-S. Tancrez, "A heuristic algorithm for solving large location-inventory problems with demand uncertainty," *European Journal of Operational Research*, vol. 259, no. 2, pp. 413–423, 2017.
- [16] A. Diabat, E. Dehghani, and A. Jabbarzadeh, "Incorporating location and inventory decisions into a supply chain design problem with uncertain demands and lead times," *Journal of Manufacturing Systems*, vol. 43, pp. 139–149, 2017.
- [17] J. Shu, Q. Ma, and S. Li, "Integrated location and two-echelon inventory network design under uncertainty," *Annals of Operations Research*, vol. 181, pp. 233–247, 2010.
- [18] M. Farahani, H. Shavandi, and D. Rahmani, "A location-inventory model considering a strategy to mitigate disruption risk in supply chain by substitutable products," *Computers & Industrial Engineering*, vol. 108, pp. 213–224, 2017.
- [19] A. Diabat, T. Abdallah, and A. Henschel, "A closed-loop location-inventory problem with spare parts consideration," *Computers & Operations Research*, vol. 54, pp. 245–256, 2015.
- [20] J. Asl-Najafi, B. Zahiri, A. Bozorgi-Amiri, and A. Taheri-Moghaddam, "A dynamic closed-loop location-inventory problem under disruption risk," *Computers & Industrial Engineering*, vol. 90, pp. 414–428, 2015.
- [21] O. Kaya and B. Urek, "A mixed integer nonlinear programming model and heuristic solutions for location, inventory and pricing decisions in a closed loop supply chain," *Computers & Operations Research*, vol. 65, pp. 93–103, 2016.
- [22] H. Guo, C. Li, Y. Zhang, C. Zhang, and Y. Wang, "A Nonlinear Integer Programming Model for Integrated Location, Inventory, and Routing Decisions in a Closed-Loop Supply Chain," *Complexity*, vol. 2018, pp. 1–17, 2018.
- [23] H. Guo, C. Li, Y. Zhang, C. Zhang, and M. Lu, "A Location-Inventory Problem in a Closed-Loop Supply Chain with Secondary Market Consideration," *Sustainability*, vol. 10, no. 6, p. 1891, 2018.
- [24] H. Guo, Y. Zhang, C. Zhang, Y. Liu, and Y. Zhou, "Location-inventory decisions for closed-loop supply chain management in the presence of the secondary market," *Annals of Operations Research*.

- [25] M. S. Pishvaei, R. Z. Farahani, and W. Dullaert, "A memetic algorithm for bi-objective integrated forward/reverse logistics network design," *Computers & Operations Research*, vol. 37, no. 6, pp. 1100–1112, 2010.
- [26] S. M. Hatefi and F. Jolai, "Robust and reliable forward-reverse logistics network design under demand uncertainty and facility disruptions," *Applied Mathematical Modelling*, vol. 38, no. 9-10, pp. 2630–2647, 2014.
- [27] S. Axsäter, "Using the deterministic EOQ formula in stochastic inventory control," *Management Science*, vol. 42, no. 6, pp. 830–834, 1996.
- [28] G. D. Eppen, "Note—Effects of Centralization on Expected Costs in a Multi-Location Newsboy Problem," *Management Science*, vol. 25, no. 5, pp. 498–501, 1979.
- [29] Z.-J. M. Shen, "Integrated supply chain design models: a survey and future research directions," *Journal of Industrial and Management Optimization*, vol. 3, no. 1, pp. 1–27, 2007.
- [30] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [31] B. Alatas, E. Akin, and A. Karci, "MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules," *Applied Soft Computing*, vol. 8, no. 1, pp. 646–656, 2008.
- [32] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

Research Article

Simulation-Based Optimization on the System-of-Systems Model via Model Transformation and Genetic Algorithm: A Case Study of Network-Centric Warfare

Bong Gu Kang ¹, Seon Han Choi ², Se Jung Kwon,³ Jun Hee Lee ⁴, and Tag Gon Kim ⁴

¹*KT Institute of Convergence Technology, Seoul, Republic of Korea*

²*Korea Institute of Industrial Technology (KITECH), Ansan, Republic of Korea*

³*Samsung Electronics Co. Ltd., Seoul, Republic of Korea*

⁴*School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea*

Correspondence should be addressed to Seon Han Choi; gigohan01@kaist.ac.kr

Received 29 December 2017; Accepted 18 July 2018; Published 3 October 2018

Academic Editor: Laszlo T. Koczy

Copyright © 2018 Bong Gu Kang 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.

Simulation of a system-of-systems (SoS) model, which consists of a combat model and a network model, has been used to analyze the performance of network-centric warfare in detail. However, finding the combat model parameters satisfying the required combat power using simulation can take a long time for two reasons: (1) the prolonged execution time per simulation run and (2) the enormous number of simulation runs. This paper proposes a simulation-based optimization method for the SoS-based simulation model to overcome these problems. The method consists of two processes: (1) the transformation of the SoS-based model into an integrated model using the neural network to reduce the execution time and (2) the optimization of the integrated model using the genetic algorithm with ranking and selection to decrease the number of simulation runs. The experimental result reveals that the proposed method significantly reduced the time for finding the optimal combat parameters with an acceptable level of accuracy.

1. Introduction

In modern warfare, communication, responsible for the flow of information between combat entities, is a primary factor in attaining victory. That is, without proper communication, the combat commander cannot obtain detection information and forward appropriate attack orders to combat entities. Such a battlefield environment based on communication systems is called network-centric warfare (NCW) [1, 2]. Many researchers in the defense modeling and simulation (DM&S) domain have recognized the importance of NCW and have tried to analyze combat systems using war game simulators reflecting the effects of communication. In earlier studies, researchers depicted communication effects by using a simplified model or by assuming perfect communication in a combat model from the perspective of a standalone system

[3, 4]. However, as the importance of communication and the need for a detailed depiction of the network system increase, recent studies have separated the network model from the combat model [5, 6]. Namely, they have implemented each model with dedicated modeling development tools and have interconnected the model through an interoperation standard, such as the high-level architecture (HLA) or the test and training enabling architecture from the perspective of the system of systems (SoS) [7–10].

In the SoS, the system means a set of interrelated components working together toward some common objective or purpose, and the SoS means a set of systems for a task that none of the systems can accomplish on its own [11, 12]. This SoS-based approach has a merit in that a collection of dedicated systems offers more functionality and performance than simply the sum of the constituent

systems and the conduct of various analyses by using existing models developed using the specialized tools in each domain [13–15]. For example, the combat model, which was only able to analyze the combat power, can measure the combat power under the communication environment by combining with a network model, which only can describe the communication environment.

The SoS-based NCW simulation offers the advantage of enabling the conduct of the “what-if” analysis of the combat power against various combat scenarios with related parameters in the combat model, by reflecting detailed communication effects from the network model [16, 17]. However, the high complexity of the network model and the interface connecting the two models paradoxically increase the execution time per simulation trial [18–20]. Porche et al. raised the simulation overhead of the highly complex network model and then analyzed the combat system with the metamodel of the network model [21]. Miner et al. also brought up the prolonged execution time due to the network model when developing the SoS analysis toolset, and they constructed its metamodel [19]. Although some studies have conducted the abstraction of the SoS-based NCW simulation model, they have not entirely reflected the SoS property due to using data from the simulation of the standalone network model, not the SoS.

Furthermore, because the NCW simulation model is usually represented as a discrete-event simulation model with a stochastic property, a large number of replicated simulations are required to obtain a steady state of the combat power [22]. As a result, the “what-if” analysis based on the SoS-based NCW simulation becomes a time-consuming problem due to two challenges: (1) enormous total simulation runs due to a large number of experimental points and replicated simulations per point and (2) the long execution time per simulation run. When a decision-maker wants to find the parameters of the combat model satisfying the requirement operation capability (ROC) of the combat power using the SoS-based NCW simulation, a significant computational cost is required.

One way to reduce the enormous simulation runs is to apply the optimization method for the discrete-event stochastic simulation model [17, 23, 24]. However, this process also requires the considerable time because it takes a long time per run of the SoS-based simulation model. On the other hand, another approach to dealing with the long execution time per run is to transform the SoS-based simulation model into an integrated simulation model via the abstraction of the model [25–27]. Although this can reduce the time required to conduct the exploratory analysis, for example, a full search on the design space, the “what-if” analysis is also time-consuming because the integrated simulation model requires somewhat execution time [28, 29]. That is, independently using each approach to the SoS-based NCW simulation model has a limitation when it comes to solving the two drawbacks simultaneously.

This paper applies a simulation-based optimization method to the SoS-based NCW simulation model in the form of a hybrid approach to overcome the abovementioned challenges. It consists of two processes: (1) the transformation of the SoS-based NCW simulation model into the integrated simulation model via the model abstraction using

the neural network [30, 31] and (2) the simulation-based optimization using the genetic algorithm (GA) with ranking and selection (R&S) [32, 33]. The former plays a role in reducing the execution time for the NCW simulation with an acceptable error; the latter plays a role in reducing the number of simulation runs required for finding the parameters of the combat model satisfying the given combat power in the entire design space. By integrating the merits of the two approaches, this paper finally facilitates simulation-based optimization in the SoS environment.

In our case study, we measured the accuracy and efficiency of the proposed optimization method for the SoS-based NCW simulation, which has already been validated. Consequently, the experimental results show that the first process reduced the execution time 113.31 times within the 5.6248% error range and the second process decreased the total number of simulation runs 19.76 times for obtaining the optimal solution with 95% accuracy. Considering the synergy effect, the proposed method ultimately reduced the execution time 2239 times within acceptable accuracy. Finally, our study provides an alternative method for finding the optimized parameters satisfying the given performance requirements in the enormous design space in the SoS.

This study is organized as follows. Section 2 describes the background and related work. Section 3 explains the proposed method. Section 4 discusses the experimental results from the NCW simulation. Finally, Section 5 concludes this study.

2. Background

2.1. System-of-Systems Stochastic Simulation Model for NCW. The NCW system consists of combat entities that are geographically or hierarchically dispersed and that share information through communication [34]. To describe the NCW system as depicted in Figure 1, modelers have developed a combat model for the military operation and a network model for communication effects separately from the perspective of the SoS, and they have enabled the information between two heterogeneous models to be shared through the interoperation middleware.

In this structure, the network model reflects the communication effects on information delivery among entities of the combat model, which acquires the communication effects, for example, the packet delivery ratio (PDR) or end-to-end delay. The former refers to the ratio of the number of successfully delivered packets from the source to the destination node; the latter refers to the average time the packets take to arrive at the destination from the source node [35]. By reflecting these effects, the combat model simulates the combat power.

2.2. Optimization in the Stochastic Simulation Model. The concept of optimization in the stochastic simulation models can simply be depicted in Figure 2. Since the model cannot be expressed in an analytic closed form, conducting the optimization using the inverse model is impossible. Thereby, conducting repeated simulations for solving the optimization problem is inevitable [36]. To be specific, if we find an

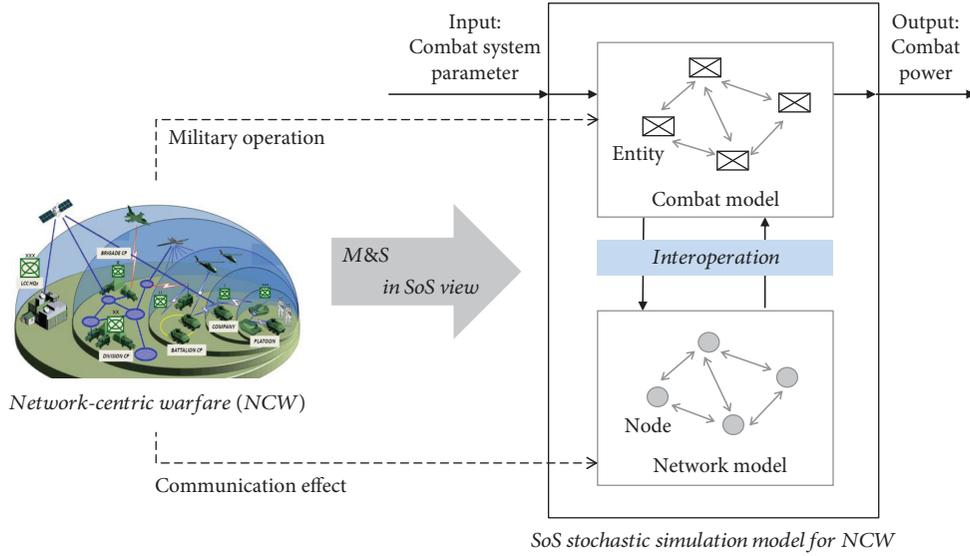


FIGURE 1: SoS-based NCW simulation model with a stochastic property [27].

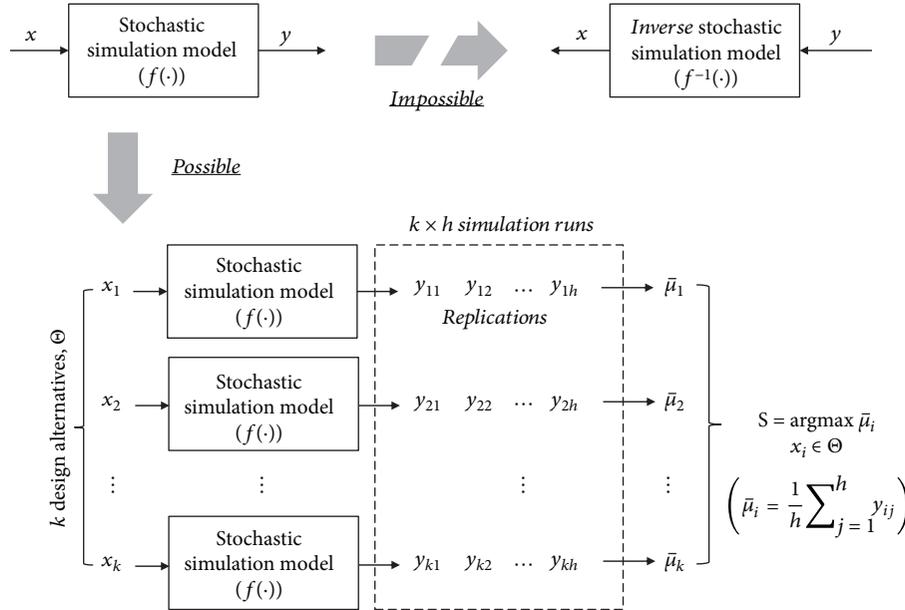


FIGURE 2: Optimization in the stochastic simulation model.

optimized parameter setting that maximizes the model's performance among k design alternatives (where design x_i refers to a particular setting of the model parameters), the total $k \times h$ simulation runs are required when considering h replications per design. To overcome the stochastic noise and to find the optimal design accurately, increasing h and obtaining the accurate performance of each design (i.e., $\bar{\mu}_i$) are necessary. However, increasing h causes efficiency problems in the optimization due to the increased cost of repeating the simulation as previously mentioned. The greater the number of design alternatives k , the worse this problem becomes.

Many methods for solving the efficiency problem have been proposed. These methods can be classified into two

categories according to the number of alternatives [37]. If k is infinite or very large, classical metaheuristic search methods, such as simulated annealing [38], GA [39], and tabu search [40], can be considered efficient approaches. These methods reduce the required number of simulation runs by performing the simulations only for selected alternatives rather than for all alternatives as depicted in Figure 3(a). To accurately find optimal designs using these methods, selecting alternatives for the simulation, including such optimal designs, is important.

On the other hand, if k is finite and relatively small (specifically, all alternatives can be simulated more than five times), the ranking and selection (R&S) methods in statistics, such as OCBA [41], KN [42], and UE [43], can be considered

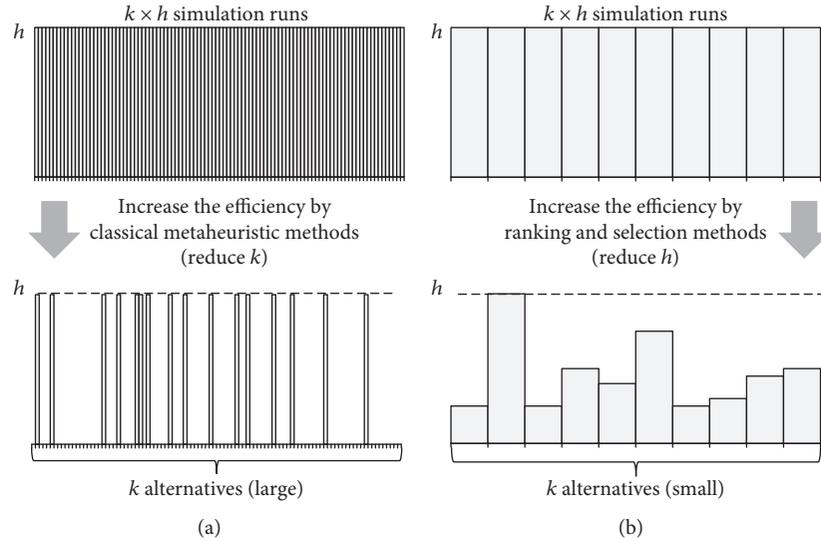


FIGURE 3: Various approaches for increasing the efficiency of the optimization in the stochastic simulation model: (a) classical metaheuristic methods and (b) R&S methods.

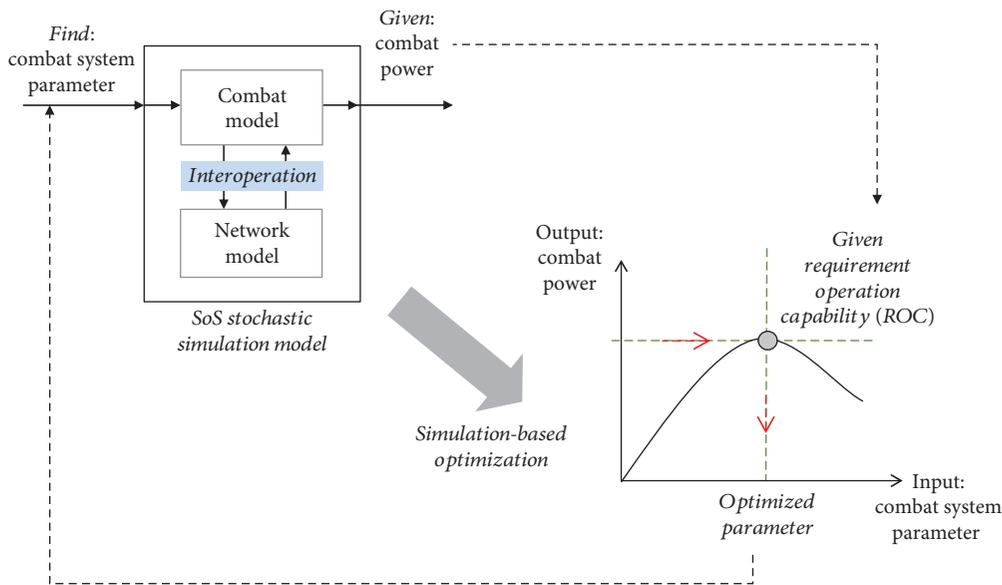


FIGURE 4: Optimization of the SoS-based NCW stochastic simulation model.

efficient approaches. Unlike classical metaheuristic methods, statistical R&S performs simulations for all alternatives. It reduces the total number of simulation runs by appropriately adjusting h in each design as indicated in Figure 3(b). That is, the aim of R&S is to select the optimal designs accurately by allocating the limited number of simulation runs effectively. Because the R&S performs the simulations for all designs, it is less likely to fall into a local optimization compared with metaheuristic methods that perform the simulations only for the selected alternatives, and it ensures that the discovered optimal design is the global optimum. However, it is clear that the R&S cannot be applied to the case of a large number of alternatives because

the simulation should be performed for all alternatives at least more than five times.

3. Proposed Method

As illustrated in Figure 4, the objective of this study is to find the optimized input parameters of the combat model, which satisfy the given ROC corresponding to the combat model's outputs (i.e., the combat power in the SoS-based simulation model). To find the parameters as efficiently as possible, this paper is aimed at minimizing the simulation cost and time due to a large number of simulation runs while retaining the accuracy. Figure 5 reveals the overall procedure of this

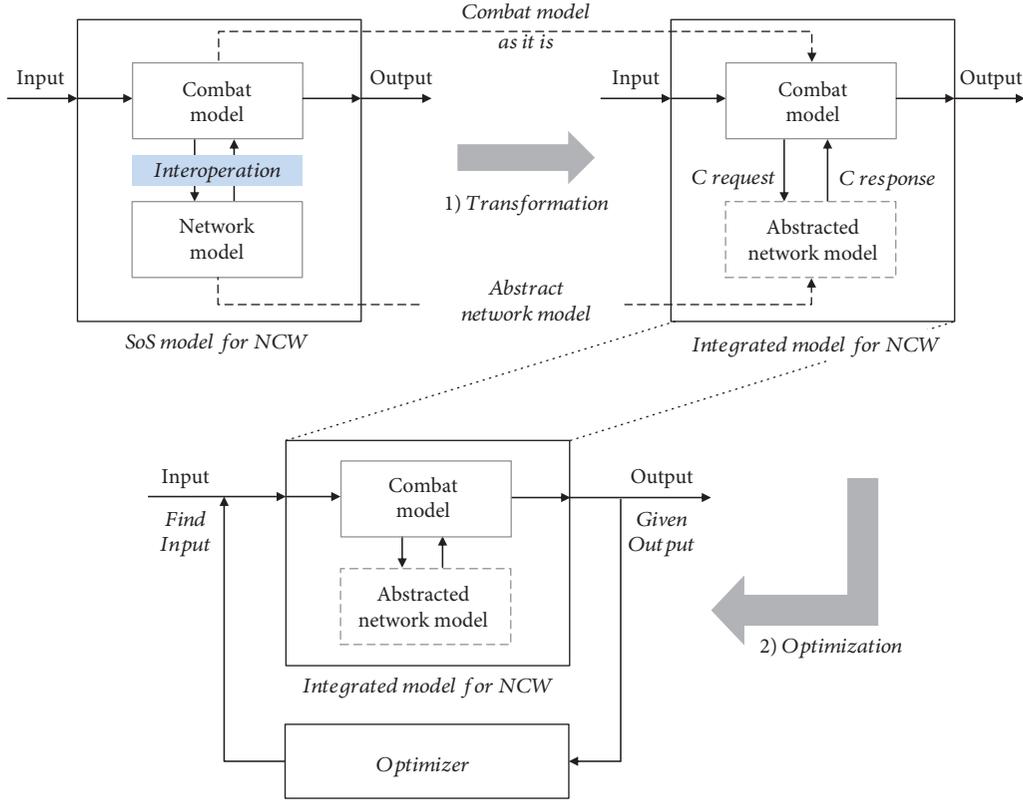


FIGURE 5: Overall procedure of the proposed method.

proposed method for minimizing the simulation cost for the optimization problem, which consists of two processes: (1) transformation and (2) optimization.

The transformation process transforms the stochastic simulation model into the integrated stochastic simulation model (upper part). For this, this paper abstracts the network model to the abstracted model. Owing to the simplified network model and the removal of the interoperation middleware, this process surely shortens the execution time per simulation run.

In the optimization process subsequently, an optimization algorithm specialized in the stochastic simulation model minimizes the number of simulation runs required for finding the optimized input in the entire design space (lower part). As a result of the two processes, the proposed method can reduce the total time for finding the parameters satisfying the required combat power. The following sections explain each process in detail.

3.1. Transformation Process. Figure 6 depicts the transformation process consisting of two phases: (1) data acquisition from the simulation of the SoS-based NCW stochastic model (right upper part) and (2) the abstracted network model construction using the neural network (right lower part). As mentioned in the prior section, the original network model plays a role in calculating the communication effects. Likewise, the abstracted model should be able to calculate the same effects. In addition, because the original network model is the discrete-event model, the abstracted network

model should also be a discrete-event model that can participate in the discrete-event simulation.

For the abstraction, data exchanged between the two models in the original SoS simulation should be collected and analyzed. From the *C response* message, the PDR and end-to-end delay can be approximated. The *C request* messages are used for calculating the interdeparture time (*intDepT*) of the messages. This paper uses the interdeparture time for the identification of the original network model [44, 45]. From the raw data, we can construct each data input and output set, $\{intDepT, PDR\}$ and $\{intDepT, DELAY\}$, for training the neural network of two communication effects.

After the data acquisition, the second phase involves constructing the abstracted network model through the model hypothesis in phase 2.1 and the variable estimation in phase 2.2. In the model hypothesis phase, a discrete-event system specification (DEVS) atomic model is constructed and will participate in the discrete-event simulation with the combat model. To estimate the PDR and end-to-end delay in the variable estimation phase, the input and output sets are trained in the nonlinear autoregressive exogenous (NARX) neural network, which is suitable for representing the dynamic and nonlinear properties of the network model [46]. Finally, this paper constructs the integrated stochastic model (left lower part of Figure 6) by substituting the network model with the identified and abstracted network model from the SoS-based NCW stochastic model (left upper part of Figure 6).

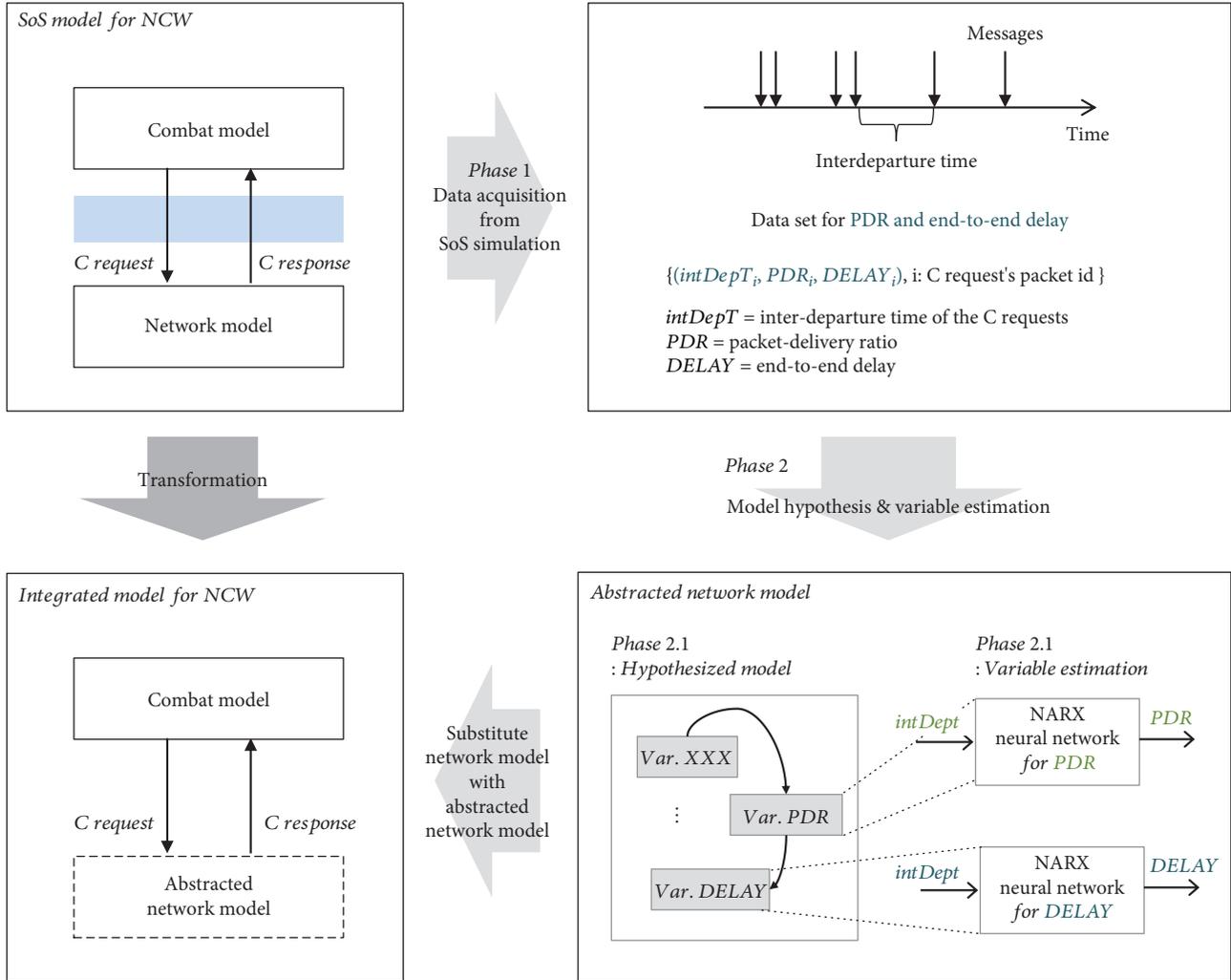


FIGURE 6: Transformation description via the model hypothesis and variable estimation [27].

Figure 7 represents the abstracted network model (ANM) based on the DEVS formalism. We hypothesize that the abstracted model, which an atomic DEVS model describes, is a probabilistic priority queue, where the $DELAY$ associated with events are considered to be priorities by influencing the P_{PDR} in generating stochastic events in the mixed time [47]. The formal specification of ANM can be found in Specification 1.

This model has three states: $(WAIT, q)$, $(COMM, q)$, and $(SEND, q)$. The first and second are the states of q without and with messages received from the combat model; the third state is the state sending the message to the combat model. When receiving a message for reflecting the communication effects in $(WAIT, q)$ from δ_{ext}^p , this model calculates $DELAY$ and P_{PDR} values using $calculateDelay$ and $calculatePDR$, which are trained via the NARX model and implemented as the functions. Then, the model stores the message with a scheduled time for output ($Cresponse$) and P_{PDR} by adding the current time and $DELAY$ in q . Then, it updates $minTA$ and transits to $(COMM, q)$. After the $minTA$ time, this model conducts

δ_{int}^p to $(SEND, q)$ based on P_{PDR} and generates the message to the combat model in $(SEND, q)$. Through such a model structure, this abstracted model can participate in the discrete-event simulation, which is impossible using only the neural network, and generates a different output according to the status of the adjacent model through the neural network.

3.2. Optimization Process. As mentioned previously, the optimization methods for the stochastic simulation model can be classified into the classical metaheuristic and R&S methods depending on the number of design alternatives k . To increase the efficiency of the optimization (i.e., to decrease the total number of simulation runs T denoted as $k \times h$ in Figure 2), the metaheuristic methods reduce the k when k is large. On the other hand, the R&S method improves the efficiency by decreasing h when k is relatively small. Although the applicable case of each method is different, the efficiency of the optimization can be further increased by reducing k and h simultaneously if both methods can be used together. Therefore,

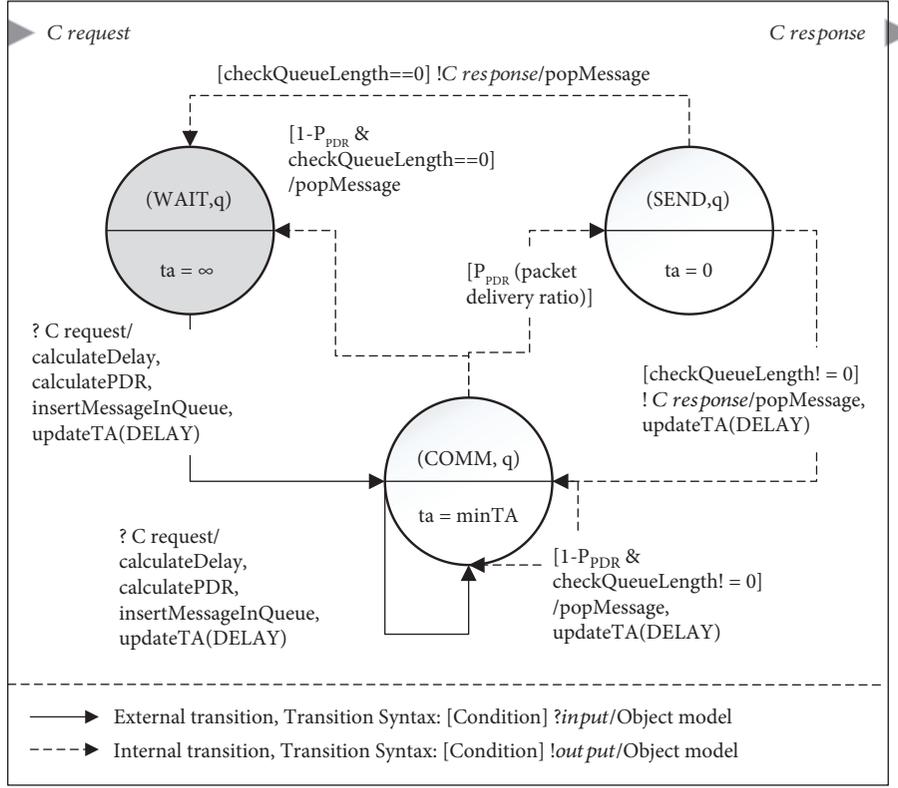
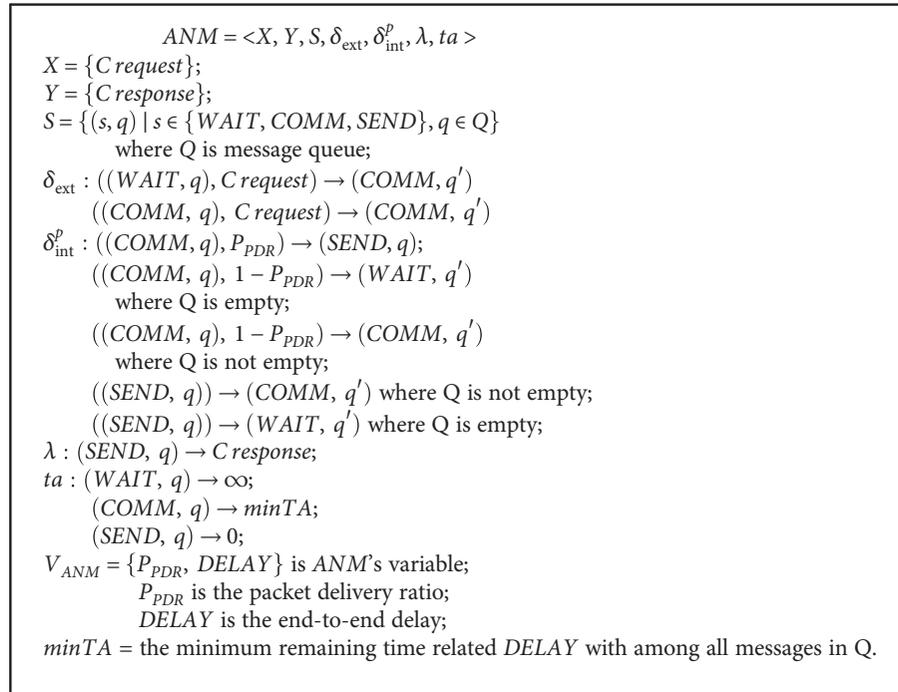


FIGURE 7: Abstracted network model using the DEVS formalism [26].



SPECIFICATION 1

in this section, we introduce a GA with R&S to improve the efficiency of the optimization for the SoS-based NCW stochastic simulation model.

The GA is a population-based search algorithm inspired by the process of natural selection [39]. Figure 8(a) represents the procedure of the GA. Each design alternative is encoded

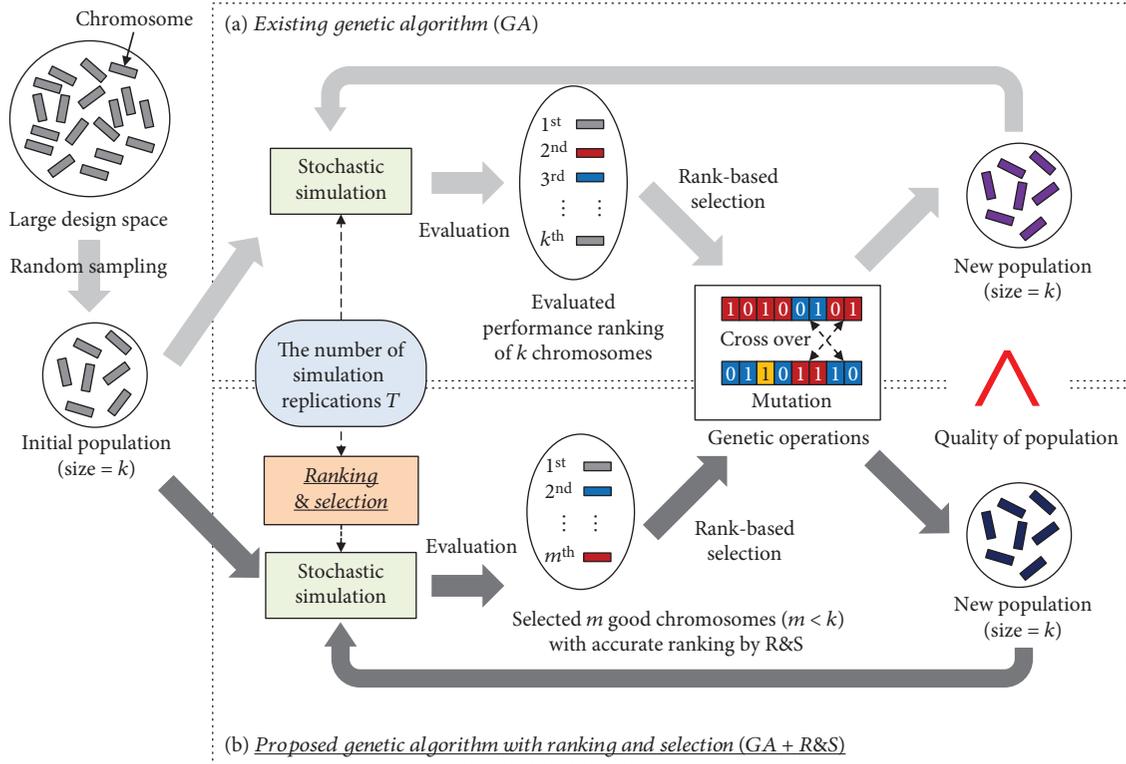


FIGURE 8: Procedures of the existing GA and the proposed GA with R&S.

as a single chromosome represented as a series of binary bits to apply genetic operations. Since the GA finds the optimal design by repeatedly updating the population through evaluating the population with stochastic simulation and applying the genetic operations, its performance highly depends on the quality of the population. That is, the more the population contains optimal or similar designs, the more the GA can find the solution efficiently with fewer repetitions.

To increase the quality of the population, two parent chromosomes selected stochastically based on the evaluated ranking revealed in Figure 8(a) should be the designs with higher performance than the other designs in the population. Accordingly, the evaluated ranking needs to be more accurate. However, estimating the exact performance of a chromosome via a large number of simulation replications for each chromosome (i.e., increasing h) reduces the GA's efficiency by increasing T . As mentioned previously, the aim of R&S is to find optimal designs accurately by allocating limited simulation replications effectively to k design alternatives. Since the size of the population is small enough to apply R&S (i.e., k is small), applying R&S to the evaluation stage of the GA can improve the quality of the population via finding the ranking accurately without increasing T as demonstrated in Figure 8(b). Consequently, the efficiency of GA can be increased [37, 48, 49].

Among various R&S algorithms, we apply the UEmr algorithm which is the most recently developed based on the uncertainty evaluation (UE) framework [43]. Algorithm 1 represents the UEmr algorithm (see [50] for detail). This algorithm focuses on finding good designs of which the

performance is within the top- m and identifying their rankings accurately. To maximize the accuracy under the limited number of simulation runs, the algorithm allocates further simulation runs Δ sequentially based on the uncertainty. The uncertainty, the criterion for allocating Δ for each design, is defined with the p value of the statistical hypothesis test. It indicates the degree to which the observed simulation results of design, such as the sample mean and standard error, can be significant evidence for the correct selection of the design. A small value of the uncertainty close to zero means that the observed results are significant evidence, so the selection of the design can be statistically considered correct. On the other hand, a relatively larger value means that the results cannot be the evidence, so it is uncertain whether the selection of the design is correct. Depending on the meaning of the uncertainty, it is necessary to allocate more simulation runs to designs with a relatively higher value of the uncertainty, as shown in step 5 of Algorithm 1.

Compared with the other similar R&S algorithms, such as UEm [23], OCBAm+ [48], OCBAm [51], and EOC- m [49], the UEmr can find the top- m designs and their rankings with higher accuracy if the simulation model has a large amount of stochastic noise. In the case of complex models, such as the SoS-based NCW model covered in this paper, stochastic noise tends to be great due to its high complexity; thus, the UEmr property of high robustness to noise allows the GA to perform the optimization of such complex models more efficiently. To verify this, we applied the UEmr and the similar R&S algorithms to the evaluation stage of the GA and compared the improved efficiency for the three kinds

Parameters: $T(\leq kn_0)$, n_0 , $\Delta(\ll T)$, $m(\leq k)$

INITIALIZATION:

- 1: **simulate** n_0 times for each design x_i , $i \in \{1, \dots, k\}$
- 2: **set** $T \leftarrow T - kn_0$

LOOP:

- 3: **while** $T \geq 0$ **do**
- 4: **calculate** the uncertainty ω_i for each x_i using the simulation results
- 5: **allocate** Δ to all designs according to ω_i :

$$a_i/a_j = \omega_i/\omega_j,$$

where $i, j \in \{1, \dots, k\}$, $i \neq j$, and $\Delta = \sum_{i=1}^k a_i$
- 6: **simulate** a_i times for each x_i additionally
- 7: **update** the simulation results
- 8: **set** $T \leftarrow T - \Delta$
- 9: **end while**

FINALIZATION:

- 10: **return** the top- m designs and their performance ranking based on the simulation results

ALGORITHM 1 [50]: Find the top- m designs out of k designs including the performance ranking of the m designs.

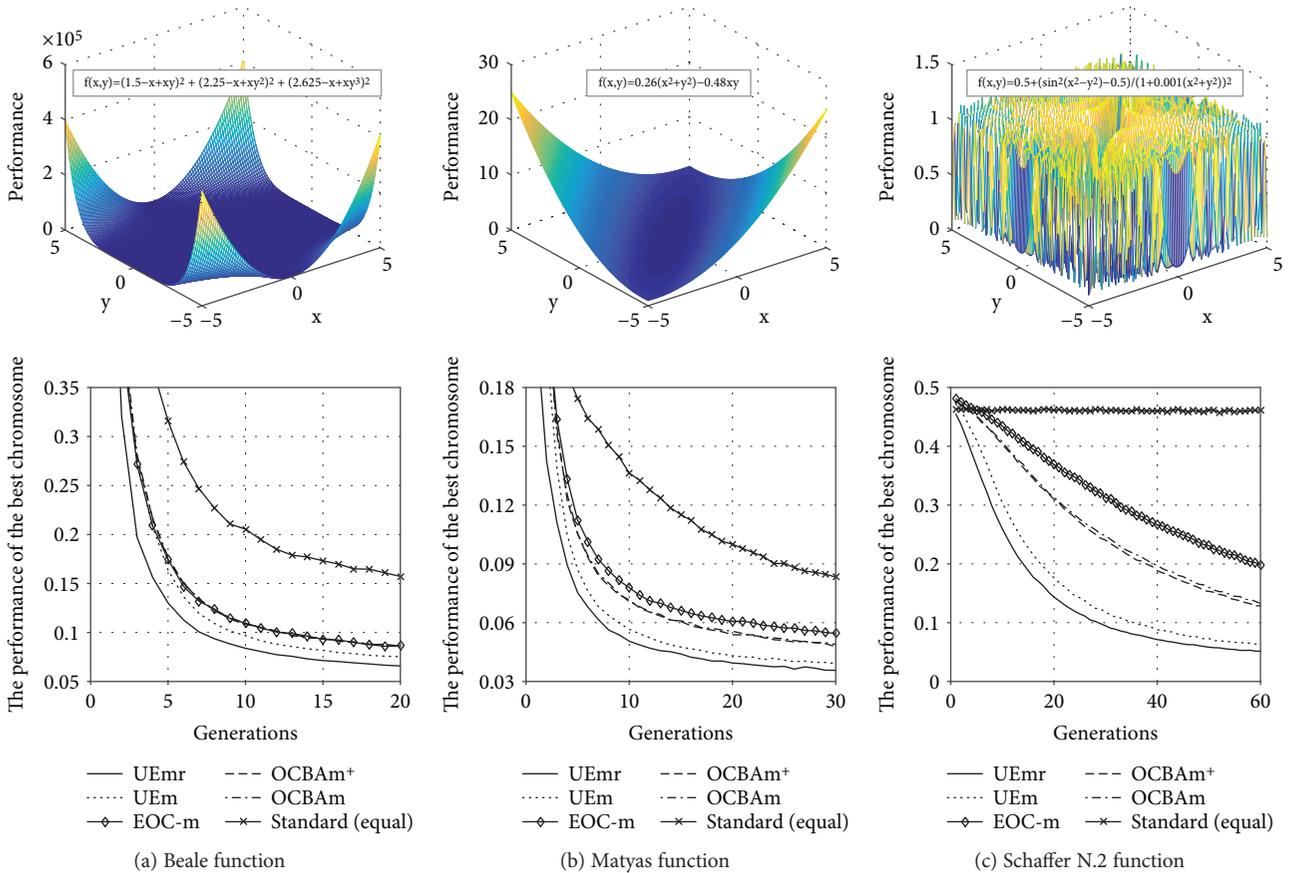


FIGURE 9: Comparative results of GA with various R&S algorithms on the three benchmark functions.

of benchmark models of which the optimal performance value is zero. The comparative results revealed in Figure 9 demonstrate the improved efficiency of the GA with the UEmr. In particular, in the case of Schaffer's model, in which the stochastic noise is relatively greater than that in the other models due to many similar peaks (where the

standard GA did not even converge), the GA with UEmr showed the highest efficiency improvement. Algorithm 2 represents the proposed GA with the UEmr.

Algorithm 2 has various parameters, and they can be classified into the two groups of GA and R&S. Among the parameters for the GA, k and G_{\max} mean the size of the

```

Parameters:
GA:  $k, G_{\max}, e(\leq m), P_c, P_m$ 
R&S (Algorithm 1):  $T(\leq kn_0), n_0, \Delta(\ll T), m(\leq k)$ 
INITIALIZATION:
1: choose  $k$  chromosomes randomly as initial population
LOOP:
2: for  $i = 1$  to  $G_{\max}$  do
    EVALUATION WITH R&S:
3: evaluate and find elite  $m$  chromosomes from the population with Algorithm 1 ( $T, n_0, \Delta, m$ )
    ELITISM:
4: put the best  $e$  chromosomes among the elite  $m$  chromosomes into the emptied new population
    GENETIC OPERATIONS:
5: do
6: choose 2 parent chromosomes from the elite  $m$  chromosomes with the linear ranking selection
7: generate 2 child chromosomes from the 2 parents with the uniform crossover with  $P_c$  rate
8: mutate the 2 children with the simple bit inversion method with  $P_m$  rate
9: put the 2 children into the new population
10: while size of the new population  $\leq k$ 
11: end for
FINALIZATION:
12: return the best chromosome among the population

```

ALGORITHM 2: Proposed GA with R&S.

population and the number of generations, respectively. P_c and P_m refer to the crossover and mutation probabilities, respectively, used in the genetic operations. Parameter e indicates the number of chromosomes for the elitism that guarantees that the quality of the population does not degrade as the generation progresses [52]. On the other hand, among the parameters for the R&S, T is the total number of simulation runs that can be allocated to the k designs in the population via Algorithm 1 from each generation. n_0 indicates the initial number of simulation runs for the k designs to obtain the minimum simulation results for further allocation (as mentioned previously, due to this n_0 , R&S cannot be applied when k is large). Δ means the additional number of replications to be distributed to the k designs according to the evaluated uncertainties in each iteration of the loop. Finally, m is the number of the top- m designs selected by Algorithm 1.

In summary, Algorithm 2, which is a combination of the GA and R&S, can further increase the efficiency of the optimization for the stochastic simulation model by reducing k and h in Figure 2 simultaneously through applying both metaheuristic and R&S methods together. That is, applying R&S to the evaluation stage of the GA can improve the quality of the population by finding the good chromosomes and their rankings accurately under limited simulation runs. The experimental results in Section 4.2 demonstrate the improved efficiency of Algorithm 2 in the optimization of the SoS-based NCW stochastic simulation model.

4. Experimental Results

The objective of this section is to demonstrate how much the proposed method reduces the simulation costs while retaining accuracy when finding the parameters satisfying

the required combat power. Here, the given required combat power is to maximize the loss-exchange ratio (LER), which refers to the ratio of the number of red casualties to the number of blue casualties. That is, the optimization problem in this section is finding the optimal parameters of NCW [53, 54] that maximize the LER using the SoS-based NCW simulation.

Figure 10 illustrates the scenario of the NCW simulation model that includes complex and hierarchical information exchange among combat entities [55, 56]. When the red force approaches, a blue force's soldier detects it and hierarchically sends the detection information to company C2 through intelligence fusion (IF). Then, C2 performs threat evaluation and weapon assignment (TEWA) and transmits the results to indirect weapons for the attack. Afterward, when the red force's combat power is below a certain level, it instructs the subordinate entities to begin close combat for the direct attack. In this situation, the information exchange (blue arrow) occurs through communication, and thereby, its performance influences the combat power.

According to this scenario, the blue force executes a defense operation against the red force with three times the military strength in a $2\text{ km} \times 2\text{ km}$ operation area. The combat model represents the army's military logic at an infantry company level including 131 entities, implemented in the DEVSimHLA [57, 58]. The network model depicts the mobile ad hoc network (MANET) based on the destination sequenced distance vector (DSDV) routing protocol including 131 nodes corresponding to the entities, implemented in the ns-3 discrete-event network simulator [59–62]. The two models participate in the SoS-based NCW simulation using HLA-RTI, which manages simulation time and exchanges the data [63]. The environment for this case study is as follows. The combat model uses CPU I5-3550

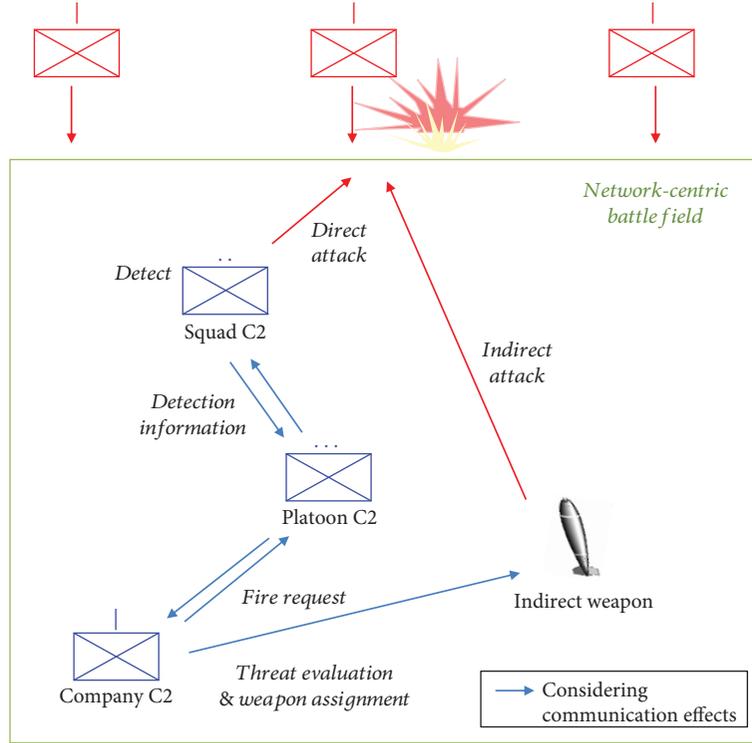


FIGURE 10: A brief description of the combat scenario of NCW.

(3.3 GHz), RAM (4 GB), and DEVSim++ v.3.1; the network model uses NS3 v.3.18. They used RTI 1.3-NG and progressed simulation time over 5000 sec. For the training, this paper uses the MATLAB neural network toolbox v.8.2.1 and MATLAB Coder v.2.7.

4.1. Experimental Design and Results: Transformation. For the first process of the proposed optimization method, we applied the transformation process of Section 3.1 to the SoS-based NCW simulation to reduce the simulation execution time. To acquire the data for the transformation, we conducted the SoS-based NCW simulation. Among the full factorial design spaces, we selected 34 experimental points for training using a Latin hypercube design [60] and conducted 30 replicated simulations against each point. According to the process in Figure 6, we constructed an abstracted network model with the NARX model, which consists of two input delays, two output delays, and 10 neurons in the hidden layer using the Levenberg-Marquardt algorithm. After the transformation, we chose an extra 17 test points at random to compare the experimental results from the SoS-based simulation with the integrated simulation with those from the replicated simulation.

From the perspective of accuracy, the absolute root-mean-square error (RMSE) of the LER indicates 0.0742, which means a relative 5.6248% error when considering the minimum and maximum values (0.8287 and 2.149, respectively). In addition, to analyze the speed, we compared the average simulation execution time and the average number of processed events per trial in the SoS-based and integrated simulation. The SoS-based simulation indicates

67.083 minutes and $3.0782e+08$ events, respectively, and the integrated simulation indicates 0.592 minutes and $2.5605e+06$ events, respectively. Assuming that the time for training is negligible, we could reduce the simulation execution time about 113.31 times by abstracting the network model with enormous events and removing the interoperation middleware.

4.2. Experimental Design and Results: Optimization. For the second process of the proposed optimization method, we applied the proposed algorithm (Algorithm 2) to the integrated model, which was developed in the previous process, to efficiently find the optimal parameter setting that maximizes the LER by reducing the number of simulation runs. A total of 16,384 design alternatives were generated depending on the combat model parameters in Table 1 (i.e., $x_i = [P_{CIF}, P_{CTE}, P_{CWA}, P_{OC}, P_{PIF}, P_{SIF}, P_{IWT}]$, and each design was encoded to a chromosome of 14 bits (i.e., two bits per parameter in Table 1). The parameters of GA in Algorithm 2 were set as follows according to the typical Dejong setting [64]: $k = 50$, $G_{\max} = 1000$, $e = 15$, $P_c = 0.5$, and $P_m = 0.01$. The parameters of R&S were set as follows according to the efficient setting guideline [43]: $T = 5000$, $n_0 = 10$, $\Delta = 50$, and $m = 30$. As a result, we could find the optimal design that maximizes the LER, [20, 20, 20, OC4, 20, 20, type 4], among 16,384 alternatives very efficiently via applying Algorithm 2. Table 2 and Figure 11 demonstrate the improved efficiency of Algorithm 2 compared to the standard GA. Here, the accuracy implies the probability of correct selection [37] and was estimated over 1000 independent trials.

TABLE 1: Combat model parameters and their description.

Parameter name	Parameter level	Description
Company C2 intelligence fusion time (P_{CIF})	20, 40, 60, 80 (sec)	The time required for intelligence fusion in company C2
Company C2 threat evaluation time (P_{CTE})	20, 40, 60, 80 (sec)	The time required for threat evaluation in company C2
Company C2 weapon assignment time (P_{CWA})	20, 40, 60, 80 (sec)	The time required for weapon assignment in company C2
Operation criteria for close combat (P_{OC})	OC 1, 2, 3, 4	The enemy's required combat power to begin the close combat
Platoon C2 intelligence fusion time (P_{PIF})	20, 40, 60, 80 (sec)	The time required for intelligence fusion in platoon C2
Squad C2 intelligence fusion time (P_{SIF})	20, 40, 60, 80 (sec)	The time required for intelligence fusion in squad C2
Indirect weapon type (P_{IWT})	Types 1, 2, 3, 4	The types of indirect weapons

TABLE 2: The number of simulation runs required to achieve the specific accuracy for the standard GA and Algorithm 2.

Accuracy	Standard GA	Algo. 2 (GA + R&S)	Ratio
0.8	1,350,000	85,000	15.88
0.85	1,565,000	90,000	17.39
0.9	1,835,000	110,000	16.68
0.95	2,470,000	125,000	19.76
0.99	4,000,000	215,000	18.60

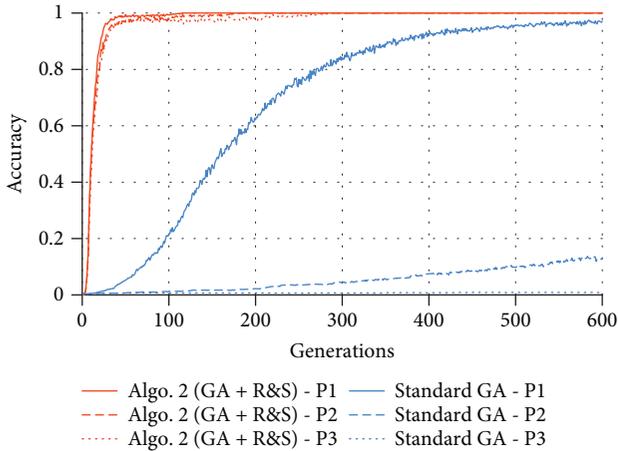


FIGURE 11: The accuracy curves of Algorithm 2 and the standard GA according to the three different settings of the algorithm parameters: P1, P2, and P3.

Algorithm 2 could greatly reduce the total number of simulation runs required to find the optimal design. As shown in Table 2, to find the optimal design with 95% accuracy, the standard GA required 2,470,000 simulation runs (5000×494 generations), whereas Algorithm 2 used only 125,000 (5000×25 generations) runs. In the case of applying only R&S instead of GA to find the optimal design, it is inefficient compared with Algorithm 2 because 163,840 simulation runs are required even if just 10 initial replications per design are conducted for 16,384 designs. Namely, Algorithm 2, which decreases both k and h by combining GA with R&S, can perform the optimization on the integrated model more efficiently than GA and R&S that only reduce k or h .

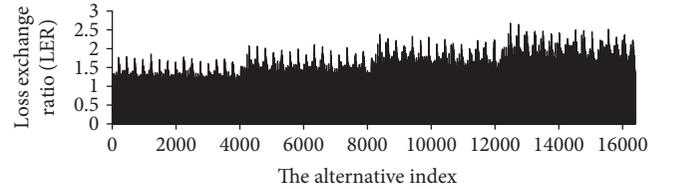


FIGURE 12: The precise LER for 16,384 designs estimated by 200 replications per design using the integrated simulation model.

As mentioned previously, the improved efficiency of Algorithm 2 is due to the increased quality of the population via applying R&S that finds the good chromosomes and their rankings accurately under limited simulation runs. This effect can be more clearly identified by changing the parameter settings of the algorithm, as shown in Figure 11. In Figure 11, the setting P1 means the previously mentioned setting, whereas P2 and P3 are the settings in which T and e were changed from P1 (i.e., P1[$T = 5000, e = 15$], P2[$T = 2500, e = 10$], and P3[$T = 1500, e = 5$]). Here, T and e are the parameters that have the greatest effect on the quality of the population. In the case of the standard GA, as T and e decrease, the quality of the population becomes poor due to the inaccurately evaluated rankings of chromosomes and the reduced effects of the elitism; thus, the efficiency is reduced greatly, as shown in Figure 11. On the other hand, the efficiency of Algorithm 2 does not decrease significantly because the algorithm finds relatively accurate ranking and good chromosomes via R&S. In addition, in the case where the stochastic noise is large due to many similar peaks as shown in Figure 12, the high robustness to noise of the applied UEmr further increases the efficiency of Algorithm 2.

5. Conclusion

To solve the time-consuming problem occurring when trying to find the combat model parameters satisfying the ROC, for example, combat power, this paper has proposed a simulation-based optimization method for the SoS-based NCW stochastic simulation model in the form of a hybrid approach with two processes.

In the first process, to shorten the prolonged execution time per simulation run, the SoS-based model was transformed into an integrated model by abstracting the highly

complex network model and removing the interoperation middleware. The transformation based on the neural network achieved a small level of error by maintaining the network model's characteristics. In the second process, to decrease the number of simulation runs, the simulation-based optimization for the integrated model was conducted using the GA with R&S. The simulation-based optimization significantly reduced the number of simulation runs for finding the optimized parameters compared with the optimization using the GA or R&S alone.

The empirical analysis revealed that the first process reduced the simulation execution time by 113.31 times within a 5.6248% error range; the second process decreased the number of simulation runs by 19.76 times to find the optimal parameters with 95% accuracy. Taken together, the proposed method ultimately reduced the total time by 2239 times with acceptable accuracy. We expect that the proposed method will help to conduct the simulation-based optimization on the SoS-based model in various domains, such as the military domain.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Brain Korea 21 Plus program through the National Research Foundation (NRF) of Korea funded by the Ministry of Education. The authors would like to thank the Editor, Prof. László T. Kóczy, and anonymous reviewers for their valuable comments and constructive suggestions.

References

- [1] J. Moffat, *Complexity Theory and Network Centric Warfare*, DIANE Publishing, 2010.
- [2] G. Neuneck, "The revolution in military affairs: its driving forces, elements, and complexity," *Complexity*, vol. 14, no. 1, pp. 50–61, 2008.
- [3] J. H. Kim, I.-C. Moon, and T. G. Kim, "New insight into doctrine via simulation interoperation of heterogeneous levels of models in battle experimentation," *Simulation*, vol. 88, no. 6, pp. 649–667, 2012.
- [4] A. Yang, H. A. Abbass, and R. Sarker, "Land combat scenario planning: a multiobjective approach," in *Simulated Evolution and Learning*, Lecture Notes in Computer Science, pp. 837–844, Springer, Berlin, Heidelberg, 2006.
- [5] A. Tolk, Ed., *Engineering Principles of Combat Modeling and Distributed Simulation*, John Wiley & Sons, 2012.
- [6] B. G. Kang and T. G. Kim, "Reconfigurable C3 simulation framework: interoperation between C2 and communication simulators," in *2013 Winter Simulations Conference (WSC)*, pp. 2819–2830, Washington, DC, USA, December 2013.
- [7] HLA Working Group, "IEEE standard for modeling and simulation (M&S) high-level architecture (HLA)-framework and rules," IEEE Standard, 2000.
- [8] US DoD, "TENA-the test and training enabling architecture," <https://www.tena-sda.org>.
- [9] C. E. Maldonado and N. A. Gómez Cruz, "The complexification of engineering," *Complexity*, vol. 17, no. 4, pp. 8–15, 2012.
- [10] K. H. Lee, J. H. Hong, and T. G. Kim, "System of systems approach to formal modeling of CPS for simulation-based analysis," *ETRI Journal*, vol. 37, no. 1, pp. 175–185, 2015.
- [11] B. S. Blanchard, W. J. Fabrycky, and W. J. Fabrycky, *Systems Engineering and Analysis*, Prentice Hall, Englewood Cliffs, NJ, USA, 1990.
- [12] ISO, *IEC/IEEE 15288: 2015, Systems and Software Engineering-Content of Systems and Software Life Cycle Process Information Products (Documentation)*, International Organization for Standardization/International Electrotechnical Commission, 2015.
- [13] M. Jamshidi, *Systems of Systems Engineering Principles and Applications*, CRC Press, 2008.
- [14] M. Jamshidi, *System of Systems Engineering Innovations for the Twenty-First Century*, Wiley, 2011.
- [15] S. W. Popper, S. C. Bankes, R. Callaway, and D. DeLaurentis, *System of Systems Symposium: Report on a Summer Conversation*, Potomac Institute for Policy Studies, 2004.
- [16] S. H. Choi, K.-M. Seo, and T. G. Kim, "Accelerated simulation of discrete event dynamic systems via a multi-fidelity modeling framework," *Applied Sciences*, vol. 7, no. 10, article 1056, 2017.
- [17] J. H. Hong, K. M. Seo, and T. G. Kim, "Simulation-based optimization for design parameter exploration in hybrid system: a defense system example," *Simulation*, vol. 89, no. 3, pp. 362–380, 2013.
- [18] I. Porche, L. Jamison, and T. Herbert, *Framework for Measuring the Impact of CAISR Technologies and Concepts on Warfighter Effectiveness Using High Resolution Simulation*, Rand Arroyo Center, Santa Monica CA, USA, 2004.
- [19] N. E. Miner, B. P. Van Leeuwen, K. M. Welch et al., "Evaluating communications system performance effects at a system of systems level," in *MILCOM 2012 - 2012 IEEE Military Communications Conference*, pp. 1–6, Orlando, FL, USA, October–November 2012.
- [20] B. D. Paz and J. A. Baer, "Communication effect server integration with OneSAF for mission level simulation," in *Proceedings of the 2008 Fall Simulation Interoperability Workshop, FIW 2008*, pp. 271–276, USA, September 2008.
- [21] I. Porche, R. Isaac, and W. Bradley, *The Impact of Network Performance on Warfighter Effectiveness*, Rand Arroyo Center, Santa Monica CA, USA, 2006.
- [22] A. M. Law, *Simulation Modeling and Analysis*, McGraw-Hill, 4th edition, 2007.
- [23] S. H. Choi and T. G. Kim, "Optimal subset selection of stochastic model using statistical hypothesis test," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 557–564, 2018.
- [24] S. H. Choi and T. G. Kim, "Pareto set selection for multiobjective stochastic simulation model," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [25] B. G. Kang, B. S. Kim, and T. G. Kim, "Abstraction on network model under interoperable simulation environment," in *ECMS 2016 Proceedings edited by Thorsten Claus, Frank Herrmann, Michael Manitz, Oliver Rose*, pp. 460–466, Germany, May 2016.
- [26] B. G. Kang, K. M. Seo, and T. G. Kim, "Machine learning-based discrete event dynamic surrogate model of communication system for simulating command, control, and

- communication system of systems,” *Simulation: Transactions of the Society for Modeling and Simulation International*, 2018.
- [27] B. G. Kang, K. M. Seo, and T. G. Kim, “Communication analysis of network-centric warfare via transformation of system of systems model into integrated system model using neural network,” *Complexity*, vol. 2018, Article ID 6201356, 16 pages, 2018.
- [28] J. H. Bigelow and P. K. Davis, *Implications for Model Validation of Multiresolution, Multiperspective Modeling (mRMPM) and Exploratory Analysis*, Rand Corp, Santa Monica CA, USA, 2003.
- [29] A. Brooks, S. Bankes, and B. Bennett, *Weapon Mix and Exploratory Analysis a Case Study*, Rand Corp, Santa Monica CA, USA, 1997.
- [30] Z. Xiao, L. Peng, Y. Chen, H. Liu, J. Wang, and Y. Nie, “The dissolved oxygen prediction method based on neural network,” *Complexity*, vol. 2017, Article ID 4967870, 6 pages, 2017.
- [31] T. Gligorijević, Z. Ševarac, B. Milovanović et al., “Follow-up and risk assessment in patients with myocardial infarction using artificial neural networks,” *Complexity*, vol. 2017, Article ID 8953083, 8 pages, 2017.
- [32] Y. Zhou, L. Zhou, Y. Wang, Z. Yang, and J. Wu, “Application of multiple-population genetic algorithm in optimizing the train-set circulation plan problem,” *Complexity*, vol. 2017, Article ID 3717654, 14 pages, 2017.
- [33] A. Jain and N. S. Chaudhari, “An improved genetic algorithm for developing deterministic OTP key generator,” *Complexity*, vol. 2017, Article ID 7436709, 17 pages, 2017.
- [34] D. S. Alberts, J. J. Garstka, and F. P. Stein, “Network centric warfare: developing and leveraging information superiority,” Assistant Secretary of Defense (C3I/Command Control Research Program), Washington DC, USA, 2000.
- [35] M. F. Khan, E. A. Felemban, S. Qaisar, and S. Ali, “Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (WSNs),” in *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 324–329, Dalian, China, December 2013.
- [36] M. C. Fu, “Feature article: optimization for simulation: theory vs. practice,” *Inform Journal on Computing*, vol. 14, no. 3, pp. 192–215, 2002.
- [37] C. H. Chen and L. H. Lee, “Stochastic simulation optimization: an optimal computing budget allocation,” *World Scientific*, vol. 1, 2011.
- [38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [39] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, USA, 1975.
- [40] F. Glover, “Tabu search, part I,” *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.
- [41] C. H. Chen, J. Lin, E. Yücesan, and S. E. Chick, “Simulation budget allocation for further enhancing the efficiency of ordinal optimization,” *Discrete Event Dynamic Systems*, vol. 10, no. 3, pp. 251–270, 2000.
- [42] S. H. Kim and B. L. Nelson, “On the asymptotic validity of fully sequential selection procedures for steady-state simulation,” *Operations Research*, vol. 54, no. 3, pp. 475–488, 2006.
- [43] S. H. Choi and T. G. Kim, “Efficient ranking and selection for stochastic simulation model based on hypothesis test,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1555–1565, 2018.
- [44] K. Worden, C. X. Wong, U. Parlitz et al., “Identification of pre-sliding and sliding friction dynamics: grey box and black-box models,” *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 514–534, 2007.
- [45] P. Amani, M. Kihl, and A. Robertsson, “NARX-based multi-step ahead response time prediction for database servers,” in *2011 11th International Conference on Intelligent Systems Design and Applications*, pp. 813–818, Cordoba, Spain, November 2011.
- [46] S. Salivahanan, A. Vallavaraj, and C. Gnanapriya, *Digital Signal Processing*, McGraw-Hill, 2001.
- [47] C. M. Seo, B. P. Zeigler, D. H. Kim, and K. Duncan, “Integrating web-based simulation on IT systems with finite probabilistic DEVS,” in *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, TMS 2015*, pp. 173–180, San Diego, CA, USA, April 2015.
- [48] S. Zhang, L. H. Lee, E. P. Chew, J. Xu, and C. H. Chen, “A simulation budget allocation procedure for enhancing the efficiency of optimal subset selection,” *IEEE Transactions on Automatic Control*, vol. 61, no. 1, pp. 62–75, 2016.
- [49] S. Gao and W. Chen, “Efficient subset selection for the expected opportunity cost,” *Automatica*, vol. 59, pp. 19–26, 2015.
- [50] S. H. Choi and T. G. Kim, “A heuristic approach for selecting best-subset including ranking within the subset,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018.
- [51] C. H. Chen, D. He, M. Fu, and L. H. Lee, “Efficient simulation budget allocation for selecting an optimal subset,” *Inform Journal on Computing*, vol. 20, no. 4, pp. 579–595, 2008.
- [52] S. Baluja and R. Caruana, “Removing the genetics from the standard genetic algorithm,” in *Machine Learning Proceedings 1995: Proceedings of the Twelfth International Conference on Machine Learning*, pp. 38–46, Tahoe City, CA, USA, July 1995.
- [53] M. Phythian, *Understanding the Intelligence Cycle*, Routledge, 2013.
- [54] J. Marshall, L. M. ER, J. Tapp et al., *An Analysis of the Open Architecture Warfare System Domain Model for Surface Time Critical Targets*, Naval Postgraduate School, 2007.
- [55] A. H. Dekker, *C4ISR Architectures, Social Network Analysis and the FINC Methodology: An Experiment in Military Organisational Structure*, Defence Science and Technology Organisation, 2002.
- [56] I. C. Moon, K. M. Carley, and T. G. Kim, *Modeling and Simulating Command and Control: For Organizations Under Extreme Situations*, Springer Science & Business Media, 2013.
- [57] B. P. Zeigler, T. G. Kim, and H. Praehofer, *Theory of Modeling and Simulation Integrating Discrete Event and Continuous Complex Dynamic Systems*, Academic Press, 2000.
- [58] T. G. Kim, C. H. Sung, S.-Y. Hong et al., “DEVSim++ toolset for defense modeling and simulation and interoperability,” *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 8, no. 3, pp. 129–142, 2011.
- [59] S. Kaur, S. kaur, and C. Sharma, “An overview of mobile ad hoc network: application, challenges and comparison of routing protocols,” *IOSR Journal of Computer Engineering*, vol. 11, no. 5, pp. 7–11, 2013.
- [60] C. Rajabhushanam and A. Kathirvel, “Survey of wireless MANET application in battlefield operations,” *International*

Journal of Advanced Computer Science and Applications, vol. 2, no. 1, pp. 50–58, 2011.

- [61] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, “Network simulations with the ns-3 simulator,” in *SIGCOMM Demonstration*, p. 527, Seattle, WA, USA, August 2008.
- [62] J. Pan and R. Jain, *A Survey of Network Simulation Tools: Current Status and Future Developments*, Washington University, 2008.
- [63] M.-W. Yoo, C. Choi, and T. G. Kim, “High-level architecture service management for the interoperation of federations,” *Simulation*, vol. 91, no. 6, pp. 566–590, 2015.
- [64] K. A. DeJong and W. M. Spears, “An analysis of the interacting roles of population size and crossover in genetic algorithms,” in *Parallel Problem Solving from Nature. PPSN 1990. Lecture Notes in Computer Science*, pp. 38–47, Springer, Berlin, Heidelberg.

Research Article

A Binary Cuckoo Search Big Data Algorithm Applied to Large-Scale Crew Scheduling Problems

José García ¹, Francisco Altimiras ^{2,3}, Alvaro Peña,¹ Gino Astorga,⁴ and Oscar Peredo²

¹Escuela de Ingeniería en Construcción, Pontificia Universidad Católica de Valparaíso, 2362807 Valparaíso, Chile

²Telefónica Investigación y Desarrollo, Santiago, Chile

³Faculty of Engineering and Sciences, Universidad Adolfo Ibañez, Santiago, Chile

⁴Universidad de Valparaíso, 2361864 Valparaíso, Chile

Correspondence should be addressed to José García; joseantonio.garcia@telefonica.com

Received 27 September 2017; Accepted 9 May 2018; Published 30 July 2018

Academic Editor: Laszlo T. Koczy

Copyright © 2018 José García 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 progress of metaheuristic techniques, big data, and the Internet of things generates opportunities to performance improvements in complex industrial systems. This article explores the application of Big Data techniques in the implementation of metaheuristic algorithms with the purpose of applying it to decision-making in industrial processes. This exploration intends to evaluate the quality of the results and convergence times of the algorithm under different conditions in the number of solutions and the processing capacity. Under what conditions can we obtain acceptable results in an adequate number of iterations? In this article, we propose a cuckoo search binary algorithm using the MapReduce programming paradigm implemented in the Apache Spark tool. The algorithm is applied to different instances of the crew scheduling problem. The experiments show that the conditions for obtaining suitable results and iterations are specific to each problem and are not always satisfactory.

1. Introduction

With the increase of different kinds of electronic devices, social networks, and the Internet of Things, the datasets are growing fast in volume, variety, and complexity. Currently, big data is emerging as a trend and working with large datasets typically aimed at extracting useful knowledge from them. To address this problem, different programming models have been developed, in which MapReduce is one of the most powerful [1].

In complex industrial systems, the engineers face challenges daily where their job is to make decisions on how to improve the production and reduce costs. They are continuously selecting where, how, when, and what it must do to achieve efficiency in the processes. Normally, these decisions are based on an optimization problem. On the other hand, nowadays, a greater data quantity is available and therefore we can build robust optimization models that support these decisions. However, this increase in data volume and variety

implies an increase in the complexity of the calculations and therefore in the convergence time of the algorithms.

Moreover, computational intelligence and particularly metaheuristics have been successful in solving complex industrial problems. In the literature, we find metaheuristics that have satisfactorily solved problems of resource allocation [2, 3], vehicle routing [4], scheduling problems [5], reshuffling operations at maritime container terminals problems [6], antenna positioning problems [7], covering problems [8, 9], and also in bioinformatics problems such as protein structure prediction, molecular docking, and gene expression analysis [10]. However, in the big data era, the integration of metaheuristics into the decision-making process presents two fundamental difficulties: the first one is to get from computational intelligence algorithms, suitable results, and convergence times when dealing with large datasets, because much of the decisions must be close to real time. The second one relates to the programming model differences usually used in computational intelligence and big data algorithms.

These difficulties motivate the design and study of computational intelligence algorithms in programming models used in big data.

A recent framework in the big data area is the Apache Spark which has been widely used to solve industry problems [11]. This framework has advantages over the traditional MapReduce model, since it uses an abstraction called resilient distributed dataset (RDD). This abstraction allows to carry out operations in memory with high fault tolerance, being indicated for the use of iterative algorithms [12]. This work is mainly focused in the behavioural performance analysis of metaheuristic algorithms implemented with the big data Apache Spark tool. The specific objective is the reduction of their convergence times, to support the decision-making in complex industrial systems at the right times. For the design of the experiments, we will use the population size of the metaheuristic and the number of executors within the Apache Spark. To perform the evaluation, the average value, number of iterations, and speed up will be used. The following scenarios will be studied:

- (1) The evaluation of the average value through the variation of the solutions number.
- (2) The evaluation of iteration number through the solution number used to solve problems.
- (3) The evaluation of algorithm scalability through executor number.

These analyses aim to understand which metaheuristic algorithm conditions, related to the solutions and executors number, can obtain suitable results and times to support the decision-making process in complex industrial problems. For this study, it was decided to use the metaheuristic Cuckoo Search; however, the method presented in this article could be applied to different problems of the complex industrial systems.

Cuckoo search is a relatively new metaheuristic that currently has been widely used in solving different types of optimization problems [13]. Some examples of solved problems by the cuckoo search algorithm are the problems in satellite image segmentation [14], the resource allocation problems [3, 15], the optimal power system stabilizers design problems [16], and the optimal allocation of wind based distributed generator problems [17] among others.

In order to carry out the experiments, two types of datasets were chosen. The first one is a benchmark dataset associated to the known set covering problem and a second dataset is associated with the large-scale railway crew scheduling problems, where the number of columns fluctuates between fifty thousand and one million. The results show that adequate scalability and convergence times are not always obtained, what depends on the dataset type and the number of solutions that are being used.

The remainder of this paper is organized as follows. Section 2 briefly introduces the crew scheduling problem. Section 3 details the cuckoo search algorithm. The state of the art of binarization techniques is described in Section 4. In Section 5, we explain the Apache Spark framework. In

Sections 6 and 7, we detail the binary and distributed versions of our algorithm. The results of numerical experiments are presented in Section 8. Finally, we provide the conclusions of our work in Section 9.

2. Crew Scheduling Problems

In the crew scheduling problem (CrSP), a group of crew members is assigned to a set of scheduled trips. This allocation must be such that all trips necessarily are covered, while the safety rules and collective agreements must be respected. These allocation and restrictions make the CrSP one of the most difficult problems to solve in the transportation industry [18].

When a bibliographic search is performed, it was found that CrSP is a problem of great importance at present, appearing variations of the original problem associated mainly to the restrictions. As an example, we found CrSP applied to railway. In [19], CrSP with attendance rates was solved; a version of CrSP with fairness preferences was solved in [20]. Crew scheduling problem applications were also found for airlines and bus transportation. In a public transport of buses, in [21] a variation of CrSP was resolved. A new heuristic was proposed in [22] to solve a crew pairing problem with base constraints. In [23], a large-scale integrated fleet assignment and crew pairing problem were solved.

In this work, due to the addition of big data concepts, we will approach the CrSP in its original form. The problem is defined as follows: given a timetable of transport services which are executed every day in a certain period of hours. Each service is divided into a sequence of trips. A trip is performed by a crew, and it is characterized by a departure station, a departure time, an arrival time, and an arrival station. Given a period of time, a crew performs a roster. This is defined as a cyclical travel sequence and each roster assigns a cost.

The CrSP then consists in finding a roster subset that covers all trips, satisfying the constraints imposed and at a minimal cost. The problem is broken down into two phases:

- (1) Pairing generation: a very large number of feasible pairings is generated. A pairing is defined as a sequence of trips which can be assigned to a crew in a short working period. A pairing starts and ends in the same depot and is associated with a cost.
- (2) Pairing optimization: a selection is made of the best subset of all the generated pairings to guarantee that all the trips are covered at minimum cost. This phase follows quite a general approach, based on the solution of set-covering or set-partitioning problems.

In this research, we will assume that the pair generation phase has already been performed because we will use a benchmark dataset. Therefore, we will focus efforts in resolving the pairing optimization phase. The pairing optimization phase requires the determination of a min-cost subset of the generated pairings covering all the trips and satisfying

```

Objective function:  $f(x)$ ,  $x = (x_1, x_2, \dots, x_n)$ 
Generate an initial population of  $m$  host nests
while ( $t < \text{MaxGeneration}$ ) or (stop criterion)
  Get a cuckoo randomly (say,  $i$ ) and replace its solution by performing Lévy flights
  Evaluate its fitness  $F_i$ 
  Choose a nest among  $n$  (say,  $j$ ) randomly
  if  $F_i < F_j$  then
    Replace  $j$  by the new solution
  end if
  a fraction  $p_a$  of the worse nests are abandoned and new ones are built
  Keep the best nests
  Rank the nests and find the current best
  Pass the current best solutions to the next generation
end while

```

ALGORITHM 1: Cuckoo search algorithm.

additional constraints. Usually it is solved through the set covering problem, and depending on the specific modeled problem, it is added as some type of constraint.

The set covering problem (SCP) is well known to be NP-hard [24]. Nevertheless, different algorithms for solving it have been developed. There exist exact algorithms that generally rely on the branch-and-bound and branch-and-cut methods to obtain optimal solutions [25, 26]. These methods, however, need an effort for solving an SCP instance that grows exponential with the problem size. Then, even medium-sized problem instances often become intractable and cannot be solved anymore using exact algorithms. To overcome this issue, the use of different heuristics has been proposed [27, 28].

For example, [28] presented a number of greedy algorithms based on a Lagrangian relaxation (called the Lagrangian heuristics); Caprara et al. [29] introduced relaxation-based Lagrangian heuristics applied to the SCP. Metaheuristics have also been applied to solve SCP, some examples are genetic algorithms [30], simulated annealing [31], and ant colony optimization [32]. More recently, swarm-based metaheuristics as cat swarm [33], artificial bee colony [34], and black hole [9] were also proposed.

The SCP can be formally defined as follows. Let $A = (a_{ij})$, be a $n \times m$ zero-one matrix, where a column j cover a row i if $a_{ij} = 1$, besides a column j is associated with a nonnegative real cost c_j . Let $I = \{1, \dots, n\}$ and $J = \{1, \dots, m\}$ be the row and column set of A , respectively. The SCP consists in searching a minimum cost subset $S \subset J$ for which every row $i \in I$ is covered by at least one column $j \in S$, that is,

$$\begin{aligned}
 \min \quad & f(x) = \sum_{j=1}^m c_j x_j \\
 \text{subject to} \quad & \sum_{j=1}^m a_{ij} x_j \geq 1, \forall i \in I, \text{ and } x_j \in \{0, 1\}, \forall j \in J,
 \end{aligned} \tag{1}$$

where $x_j = 1$ if $j \in S$, $x_j = 0$ otherwise.

3. Cuckoo Search Algorithm

The cuckoo search is a bioinspired algorithm derived from some cuckoo bird species with an obligate brood parasitism, who lay their eggs in the nests of other bird species [13]. For simplicity, the cuckoo search algorithm is described using the following idealized rules:

- (1) Each cuckoo lays one egg at a time and dumps it in a randomly chosen nest.
- (2) The best nests with high-quality eggs will be carried over to the next generations.
- (3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$. In this case, the host bird can either get rid of the egg or simply abandon the nest and build a completely new nest.

The basic steps of the CS can be recapitulated as the pseudocode shown in Algorithm 1.

The updated cuckoo search solutions are shown in (2), in which γ corresponds to the step size, and \oplus corresponds to the entry-wise multiplications. A random number denominated as Levy (κ) is given by the distribution shown in (3).

$$X_{t+1} = X_t + \gamma \oplus \text{Levy}(\kappa), \tag{2}$$

$$\text{Levy}(\kappa) \sim \mu = t^{-1-\kappa}. \tag{3}$$

The search engine of the cuckoo search algorithm performs naturally in continuous spaces. Nevertheless, the crew scheduling problems are solved in discrete or binary spaces, forcing the adaptation of the original algorithm. A state of the art of main techniques used in the binarization of swarm intelligence continuous metaheuristics is presented in Section 4.

4. Binarization Methods

There exists two main categories for binarization techniques [35]. General binarization frameworks are part of one of

these groups in which exists a mechanism that allows the binary transformation of any continuous metaheuristic without altering the operators. The most used of these frameworks are the transfer functions and the angle modulation. The binarizations designed specifically for a metaheuristic are the second group of binarization methods that include techniques such as the set-based approach and the quantum binarization.

The most used binarization method is the transfer function introduced by [36]. This function is an inexpensive operator that provides the probability values and models the solution positions of the transition. The transfer function is the beginning of the binarization method that allows to map the \mathbb{R}^n solutions in $[0,1]^n$ solutions. The S shaped and the V shaped are the most used transfer functions, well described in [37, 38]. The next step is applying a rule to binarize the transfer function results, which could include the binarization rules elitist, the static probability, the complement, or the roulette [37].

The sizing optimization of the capacitor banks in radial distribution feeders was performed previously using a binary particle swarm optimization [39]. For the reliability analysis of the bulk power system, a transfer function based on swarm intelligence was used [40]. A binary coded firefly algorithm that solves the set covering problem was performed using the same transfer function [37]. A binary cuckoo search algorithm for solving the set covering problem was applied previously [41]. An improved firefly and particle swarm optimization hybrid algorithm was applied to the unit commitment problem [38]. A cryptanalytic attack on the knapsack cryptosystem was approached using the binary firefly algorithm [42]. The network and reliability constrained unit commitment problem was solved using a binary real coded firefly algorithm [43]. Similarly, using the firefly algorithm, the knapsack problem was solved [44].

The angle modulation method uses four parameters which control the frequency and shift of a trigonometric function as is shown in (4).

$$g_i(x_j) = \sin(2\pi(x_j - a_i)b_i \cos(2\pi(x_j - a_i)c_i)) + d_i. \quad (4)$$

Using a set of benchmark functions, the angle modulation method was first applied in the particle swarm optimization. Assuming a n -dimensional binary problem and $X = (x_1, x_2, \dots, x_n)$ as a solution. The first step uses a four-dimensional space, in which each dimension corresponds to a coefficient of (4). The solutions (a_i, b_i, c_i, d_i) are linked to a g_i trigonometric function. The rule 6 is used for each element x_j :

$$b_{ij} = \begin{cases} 1 & \text{if } g_i(x_j) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Now for each four-dimensional initial solution (a_i, b_i, c_i, d_i) , we obtain a feasible n -dimensional solution binarized for our n -binary problem $(b_{i1}, b_{i2}, \dots, b_{in})$. Several applications of the angle modulated method have been developed. This include the implementation of angle modulate using

a binary PSO to solve network reconfiguration problems [45]. Another implementation is a binary adaptive evolution algorithm applied to multiuser detection in multicarrier cdma wireless broadband system [46]. An angle modulate binary bat algorithm was also previously applied for the mapping of functions when handling binary problems using continuous-variable-based metaheuristics [47].

Evolutionary computing (EC) and quantum computing are two research areas involving the use of three algorithms categories [48]. First, the quantum evolutionary algorithms are focused on the application of EC algorithms in a quantum-computing environment. The evolutionary-designed quantum algorithms are focused in the automatic manufacturing of new quantum algorithms. The quantum-inspired evolutionary algorithms use some concepts and bases of quantum computing to generate new EC algorithms.

- (1) Quantum evolutionary algorithms: these algorithms focus on the application of EC algorithms in a quantum-computing environment.
- (2) Evolutionary-designed quantum algorithms: these algorithms try to automate the generation of new quantum algorithms using evolutionary algorithms.
- (3) Quantum-inspired evolutionary algorithms: these algorithms concentrate on the generation of new EC algorithms using some concepts and principles of quantum computing.

The quantum binary approach is part of this last category, in which the algorithms are adapted to be used on normal computers, integrating the concepts of q-bits and superposition. In this method, each achievable solution has a position $X = (x_1, x_2, \dots, x_n)$ and the quantum q-bits vector $Q = [Q_1, Q_2, \dots, Q_n]$. Q stands for the probability of x_j take the value 1. For each dimension j , a random number between $[0,1]$ is obtained and compared with Q_j , if $\text{rand} < Q_j$, then $x_j = 1$, else $x_j = 0$. The mechanism of Q vector updating is distinct to each metaheuristic.

The application of quantum swarm optimization has been used in different problems including combinatorial optimization [49], cooperative approach [50], knapsack problem [51], and power quality monitoring in [52]. The application of quantum differential evolution is also observed in the knapsack problem [53], combinatorial problems [54], and methods of image thresholding [55]. A quantum algorithm using cuckoo search metaheuristic was applied to the knapsack problem [56] and bin packing problem [57]. An application to image thresholding using quantum ant colony optimization is reported in [55]. Two quantum binarization applications to the knapsack problem are reported previously using harmony search in [58] and monkey algorithm in [59]. The quantum differential evolution algorithm was applied to the knapsack problem in [53], combinatorial problems [54], and image threshold methods in [55]. Using the cuckoo search metaheuristic, a quantum algorithm was applied to the knapsack problem [56] and bin packing problem [57]. A

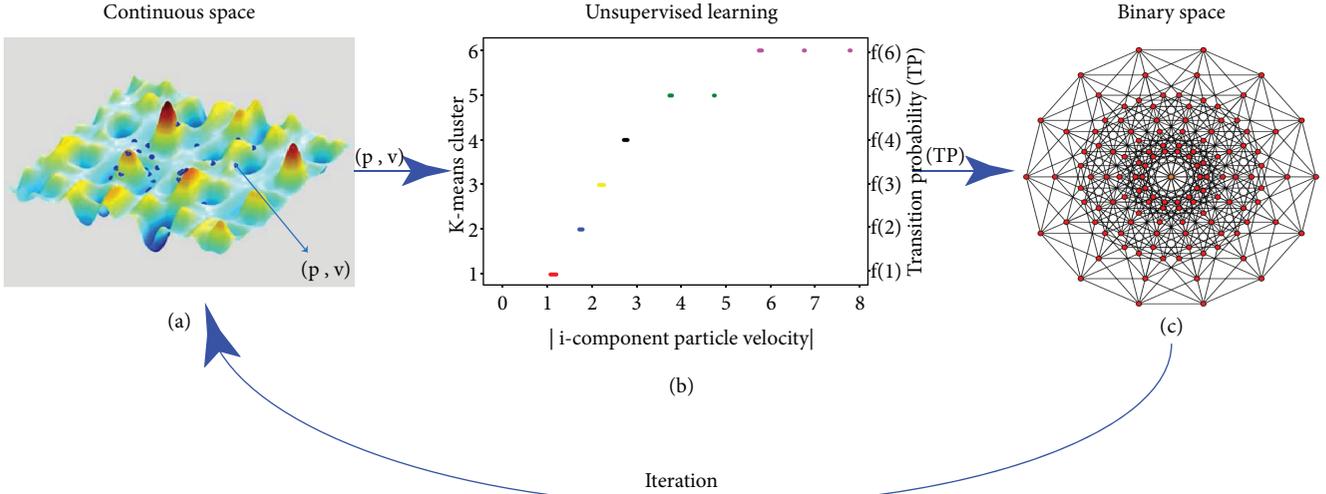


FIGURE 1: K-means binarization method.

quantum ant colony optimization was applied to image threshold in [55]. Using Harmony Search in [58] and Monkey algorithm in [59], quantum binarizations were applied to the knapsack problem.

The unsupervised learning K-means clustering method is used to perform binarization in different problems as is shown in Figure 1. This method starts with the cuckoo search algorithm generating the pair (p, v) in a continuous space, in which p is the position and v the velocity of the solution (Figure 1(a)). All the velocity module elements are considered, and the K-means is applied (Figure 1(b)). For each cluster k , we link a value $f(k) \in [0, 1]$ of the transition probability (Figure 1(b)). Finally, the transition is performed using the (6). In this equation, $\hat{x}^i(t)$ corresponds to the complement of $x^i(t) \in 0, 1$. These transitions occur in the binary space (right panel). If $x^i \in k$ cluster, then $TP(x^i) = f(k)$.

$$x^i(t+1) := \begin{cases} \hat{x}^i(t), & \text{if } \text{rand} < TP(x^i), \\ x^i(t), & \text{otherwise.} \end{cases} \quad (6)$$

In a previous work, we solved the knapsack problem by applying the transition probability function shown in (7) [3]. In this equation, $\alpha = 0.1$, $\beta = 1$, and $N(x^i)$ corresponds to the cluster that belongs x^i . P_{tr} corresponds to the transition probability; N can take values between $\{0, \dots, 4\}$. The initial probability is run by α , and then β carries on the probability jump between the different groups.

$$P_{tr}(x^i) = \alpha + \beta N(x^i) \alpha. \quad (7)$$

5. Spark Distributed Framework

The purpose of this section is to present the Spark distributed framework that it has a target to work with big volumes of data. This framework will be used later in Section 7.

The Spark framework provides an interface of friendly work that allows using of the good way the storage, the memory, and CPU and a set of servers that have as their

purpose processing large amounts of data in memory [11]. The requirement of processing large amounts of data is a need that is expressed in the last time, given principally by the low that has shown the cost of data storage which leads to a new need that is to obtain knowledge of this information gathered across the time. This new need arising out of the available storage capacity allowed to find a new line of action for researchers, since the amount, diversity, and complexity of the data [60–64], they are not capable of being tackled by the traditional methods of automatic learning.

Spark has a high performance in parallel computing, being used in machine learning algorithms [65], imaging processing [66], bioinformatics [67], computational intelligence [9], astronomy [68], medical information [69], and so on.

A pioneer in address the treatment of bulk data based on the principle of the locality of the data [70] was MapReduce framework [1] which has the disadvantage of being insufficient for applications that need to share information across several steps or for iterative algorithms [71]. The Spark framework has been very successful becoming a platform for generic use, such as batch processing, iterative process, interactive analysis, flow processing, automatic learning, and computer graphics.

The units of central data of Spark are the resilient distributed datasets (RDD). These units are distributed and are immutable, that is, the transformation of the RDD are RDD and abstraction of memory fault tolerant. Principally, there are two types of operations: transformations that take RDD and produce RDD and actions that take RDD and produce values. To execute Spark, there are several options of administration of cluster that can be used, from the simple independent solutions of Spark, Apache Mesos, and Hadoop YARN [72].

In our case and based on the engineering applications, we decide to use the management Hadoop YARN, being the latest implementation that uses cloud computing [73], that has the characteristic of putting at disposal large number of

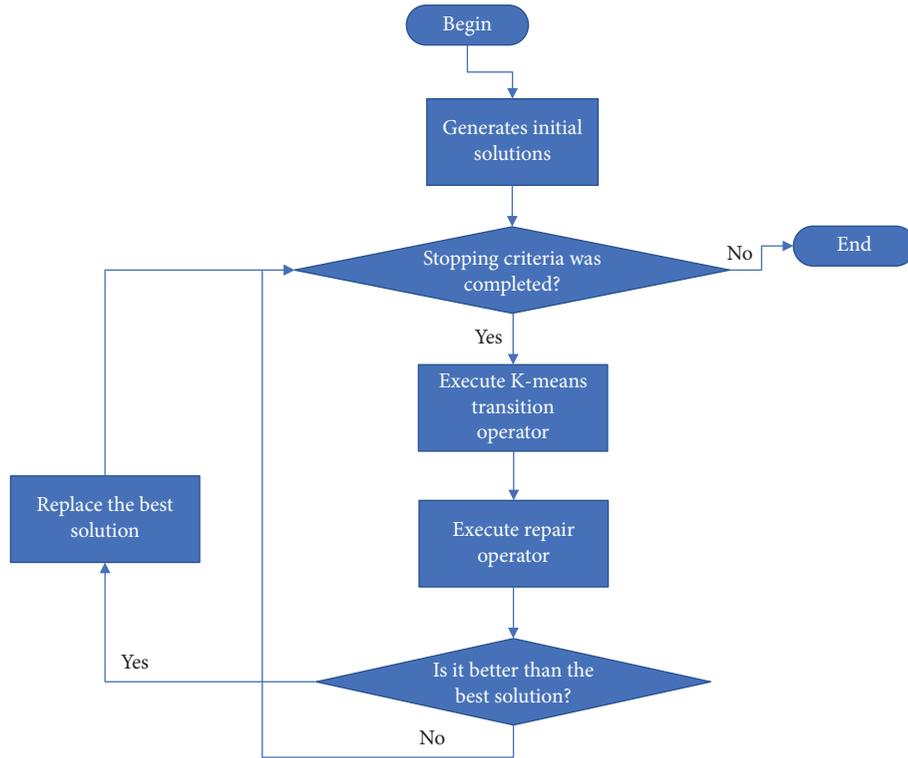


FIGURE 2: Binary cuckoo search algorithm flow chart.

devices to provide such services as computation and storage on demand that represent a lower cost of hardware, software, and maintenance [73].

6. Binary Cuckoo Search Algorithm

The general performance of the binary cuckoo search algorithm is summarized in this section. First, the algorithm creates the initial solutions with the operator. Once this happens, the algorithm evaluates compliance with the stop criterion. Maximum iteration number and obtaining the optimal value are the two stop criteria. When one of these criteria is not obtained, the K-means transition operator is executed to perform the binarization (detailed in Section 6.2). When the transitions are already obtained, a repair operator must be applied whether the solutions do not accomplish with the problem restrictions (detailed in Section 6.3). This iterative process is evaluated until the stop criterion is accomplished. A general diagram of the process described is detailed in Figure 2.

6.1. Initial Solution Operator. To obtain a new solution, the process begins with the random choice of a column. It is then queried whether the current solution covers all rows. The heuristic operator (Section 6.4) is run to add a new column, until all rows are covered, if the previous part does not happen. The final step is to delete a column, if there are columns that all their rows are covered by more than one column. The

```

1: Function Initialization()
2: Input
3: Output Initialized solution  $S_{out}$ 
4:  $S \leftarrow \text{SelecRandomColumn}()$ 
5: while All row are not covered do
6:    $S.append(\text{Heuristic}(S))$ 
7: end while
8:  $S \leftarrow \text{deleteRepeatedItem}(S)$ 
9:  $S_{out} \leftarrow S$ 
10: return  $S_{out}$ 
  
```

ALGORITHM 2: Initial solution operator.

initialization process to obtain the solution is detailed in Algorithm 2.

6.2. K-Means Transition Operator. Cuckoo search is a continuous swarm intelligence metaheuristic. The solutions position at each iteration needs to be updated due its iterative nature. This update is performed in \mathbb{R}^n space when the metaheuristic is continuous. The solution position update can be expressed in a general form for any continuous metaheuristics as is shown in (8). In this equation, $x(t+1)$ corresponds to the x position of the solution at time $t+1$. This position is obtained from the position x at time t plus a Δ function calculated at time $t+1$. The function Δ is due to each metaheuristic and generates values in \mathbb{R}^n . In cuckoo

```

1: Function K-meansTransition(ListX (t))
2: Input List solutions t (ListX (t))
3: Output List solution t + 1 (ListX (t + 1))
4:  $\Delta^i$  List  $\leftarrow$  get $\Delta^i$  (ListX (t), MH)
5:  $X^i$  Groups  $\leftarrow$  K-means ( $\Delta^i$  List, K)
6: for X(t) in ListX (t)
7:   for  $X^i$  (t) in X(t)
8:      $X^i$  Groups  $\leftarrow$  get  $X^i$  Groups (i, X(t),  $X^i$  Groups)
9:      $P_{tr}(X^i(t)) \leftarrow$  getTransitionProbability( $X^i$  Group)
10:     $X^i(t+1) \leftarrow$  applyTransitionRule( $P_{tr}(X^i(t))$ )
11:   end for
12: end for
13: for X(t+1) in ListX(t+1)
14:   X(t+1)  $\leftarrow$  Repair(X(t+1))
15: end for
16: return ListX(t+1)

```

ALGORITHM 3: K-means transition algorithm.

search, for example, $\Delta(x(t)) = \gamma \oplus \text{Levy}(\kappa)(x)$, in black hole $\Delta(x(t)) = \text{rand} \times (x_{bh}(t) - x(t))$ and in the firefly, bat, and PSO algorithms, Δ can be expressed in simplified form as $\Delta(x(t)) = v(x(t))$.

$$x(t+1) = x(t) + \Delta(x(t)) \quad (8)$$

The movements generated by the cuckoo search algorithm in each dimension for all solutions are considered in the K-means transition operator. $\Delta^i(x(t))$ is the displacement magnitude of the $\Delta(x(t))$ in the i th position for the solution x at time t . Using $\text{abs}(\Delta^i(x(t)))$, the magnitude of the displacement, the displacements are subsequently grouped. The K-means method is used to do this, where K represents the number of clusters used. In the final step, a generic P_{tr} function given in (9) is proposed to assign a transition probability. In this case, $\mathbb{Z}/k\mathbb{Z}$ is the group obtained when quotient \mathbb{Z} by $k\mathbb{Z}$, that is to say, $\mathbb{Z}/k\mathbb{Z} = \{0, 1, 2, \dots, k-1\}$, where each element of the group identifies each of the clusters. Since $P_{tr}(i)$ is a probability, it take values in $[0,1]$.

$$P_{tr} : \frac{\mathbb{Z}}{k} \mathbb{Z} \rightarrow [0, 1]. \quad (9)$$

Through the function P_{tr} , a transition probability is assigned to each group. We use the linear function given in (10) as a first approximation. In this equation, $N(x^i)$ corresponds to the location of the group to which $\Delta^i(x)$ belongs. The coefficient α allows defining the transition probability value for all the clusters. This increases proportional to α . For our particular case, $N(x^i) = 0$ corresponds to elements belonging to the group that has the lowest Δ^i values and therefore smaller transition probabilities will be assigned to them.

$$P_{tr}(x^i) = P_{tr}(N(x^i)) = \alpha + N(x^i)\alpha. \quad (10)$$

```

1: Function Repair( $S_{in}$ )
2: Input Input solution  $S_{in}$ 
3: Output The Repair solution  $S_{out}$ 
4:  $S \leftarrow S_{in}$ 
5: while needRepair(S) == True do
6:   S.append (Heuristic(S))
7: end while
8:  $S \leftarrow$  repeatedItem(S)
9:  $S_{out} \leftarrow S$ 
10: return  $S_{out}$ 

```

ALGORITHM 4: Repair algorithm.

```

1: Function Heuristic( $S_{in}$ )
2: Input Input solution  $S_{in}$ 
3: Output The new column  $C_{out}$ 
4: listRows  $\leftarrow$  getBestRows( $S_{in}$ , N=10)
5: listcolumnsOut  $\leftarrow$  getBestColumns(listRows, M=5)
6: columnOut  $\leftarrow$  getColumn(listcolumnsOut)
7: return columnOut

```

ALGORITHM 5: Heuristic operator.

The K-means transition operator begins with the calculation for each solution of the Δ^i (Algorithm 3). The solutions are then grouped using K-means clusterization and the Δ^i as magnitude of distance. We obtain the transition probability with the group assigned to each solution using (10). Subsequently, the transition of each solution is performed. The rule 12 for the cuckoo search is used to perform the transition, where \tilde{x}^i is the complement of x^i . In the final step, each solution is composed using the repair operator detailed in Algorithm 4.

$$x^i(t+1) := \begin{cases} \tilde{x}^i(t), & \text{if } \text{rand} < P_{tr}(x^i), \\ x^i(t), & \text{otherwise.} \end{cases} \quad (11)$$

6.3. Repair Operator. Using the K-means transition and the perturbation operators, the repair operator objective is to repair the solutions generated. The operator to perform the repairing process has as input parameter the solution S_{in} to repair and as output parameter the repaired solution S_{out} . We iteratively use the heuristic operator for the execution of the process, which specify the column that must be added. Once all the rows are covered, the deletion is applied to the columns that have all their rows covered by other columns.

6.4. Heuristic Operator. To repair the solutions that do not comply with the constraints is used the heuristic operator. The heuristic operator aims to select a new column for the cases that a solution needs to be built or repaired. The operator considers as input parameter the solution S_{in}

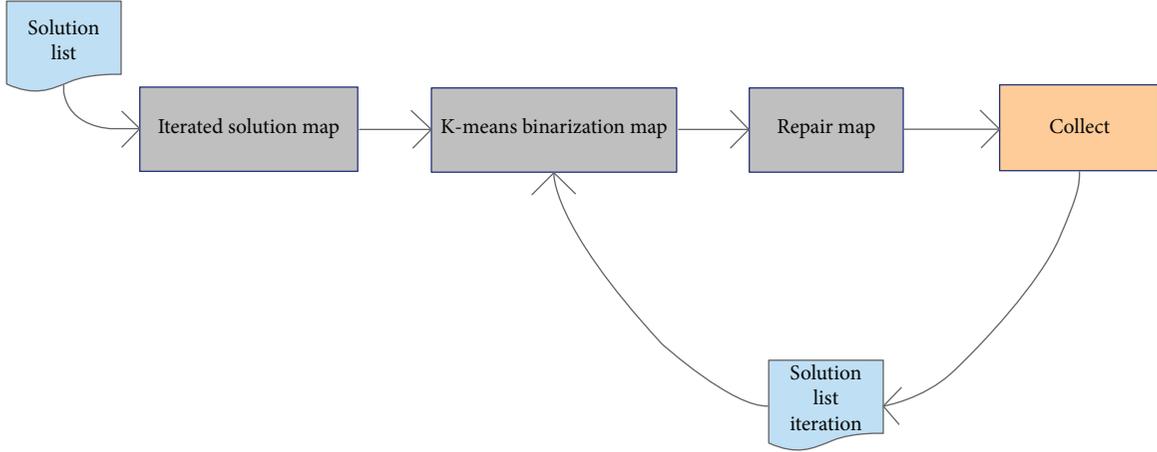


FIGURE 3: Flow chart of Spark binary cuckoo search algorithm.

```

1: Function: distributedBinaryCuckoo(ISol)
2: Input: List of solution (ISol)
3: Output: Iterated list of solution (ISol)
4: ISol ← ISol.map (lambda Solution: (idS, iteratedSolution(Sol))
5: ISol ← ISol.map (lambda Solution: (idS, K-meansTransition(Sol))
6: ISol ← ISol.map (lambda Solution: (idS, repair(Sol))
7: ISol ← ISol.collect()
8: return ISol
  
```

ALGORITHM 6: Distributed operator.

TABLE 1: SCP instances from Beasley’s OR-Library.

Instance set	n	m	Cost range	Density	Optimal solution
4	200	1000	[1100]	2	Known
5	200	2000	[1100]	2	Known
6	200	1000	[1100]	5	Known
A	300	3000	[1100]	2	Known
B	200	1000	[1100]	5	Known
C	400	4000	[1100]	2	Known
D	400	4000	[1100]	5	Known
E	500	5000	[1100]	10	Known
F	500	5000	[1100]	20	Unknown
G	1000	10000	[1100]	2	Unknown
H	1000	10000	[1100]	5	Unknown

which needs to be completed, and in the case of being a new solution $S_{in} = \emptyset$. With the list of columns belonging to S_{in} , you get the set of rows R not covered by the solution. With the set of rows not covered and using (12), we obtain in line 4 the best 10 rows to be covered. With this list of rows (*listRows*) on line 5, we obtain the list of the best columns according to the heuristic indicated in (13).

Finally, as a random process, we obtain in line 6 the column to incorporate.

$$\text{WeightRow}(i) = \frac{1}{L_i}, \quad (12)$$

Where L_i is the sum of all ones in row i ,

$$\text{WeightColumn}(j) = \frac{c_j}{|R \cap M_j|}, \quad (13)$$

Where M_j is the set of rows covered by Col j .

7. Binary Cuckoo Search Big Data Algorithm

In this section, we describe the distributed version of the algorithm developed with Apache Spark. The key in each of the map transformations and collect actions used corresponds to the solution identifier that will be denoted by *idS*. When the identifier is used as a key during the execution, it allows the calculations associated to a solution to be executed always in the same partition for the different stages and therefore to be more efficient regarding the data transfer between different workers. In Figure 3, the flow diagram for the distributed algorithm is shown, and in Algorithm 6, the pseudo-code of an iteration is detailed.

TABLE 2: Parameter setting for cuckoo search big data algorithm.

Parameters	Description	Value	Range
α	Transition probability coefficient	0.1	[0.08, 0.1, 0.12]
K	Number of transition groups or clusters	5	[4–6]
γ	Step length	0.01	[0.009, 0.01, 0.011]
κ	Levy distribution parameter	1.5	[1.4, 1.5, 1.6]
Iteration number	Maximum iterations	700	[600, 700, 800]

TABLE 3: Average result by problem type of dataset OR-Library.

Instance	Best known	BCSBA (5)	BCSBA (10)	BCSBA (20)	BCSBA (50)	BCSBA (100)	BCSBA (500)
4	510	514.1	513.3	511.6	510.9	510.9	510.9
5	257.3	259.5	258.8	258.1	257.6	257.6	257.6
6	144.2	145.1	144.9	144.7	144.5	144.5	144.5
A	241.4	243.4	243.2	243.0	242.9	242.8	242.8
B	75.2	75.4	75.2	75.2	75.2	75.2	75.2
C	224.6	227.7	226.3	226.1	225.6	225.6	225.6
D	64.2	65.3	65.1	64.9	64.7	64.7	64.6
E	28.4	28.8	28.6	28.5	28.5	28.5	28.5
F	14	14.5	14.4	14.2	14.2	14.2	14.2
G	166.4	173.2	172.4	171.6	168.1	168.1	168.0
H	59.4	64.9	63.4	63.1	60.7	60.8	60.7
Average	162.28	164.72	163.87	163.73	162.99	162.99	162.96
<i>p</i> value			2.1e-4	1.5e-5	3.5e-7	2.7e-7	3.1e-08

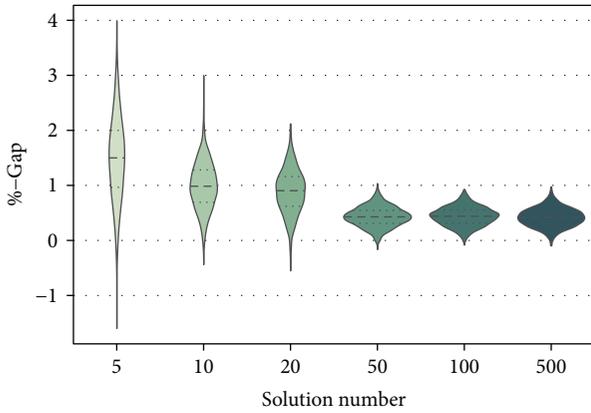


FIGURE 4: Violin chart of the results quality by solution number.

LSol contains the solution list to be iterated with the cuckoo search algorithm. Each of these solutions has the position and velocity information. The first step is to iterate the solutions using the cuckoo search algorithm; this is done at line 4. The key corresponds to the *idS* particle identifier, and the returned value corresponds to the iterated solution *Sol*, in which the velocity values have been updated. The next step is to perform an iteration of the positions. For this, the K-means transition operator described in Section 6.2 and executed at line 5 of Algorithm 6 is used. With the K-means transition operator, the velocities obtained in the previous step are used to get the new binary values of the solution position. Subsequently since there is a

possibility that the iterated solutions do not meet with the constraints, a repair operator is applied. This operator acts on the positions and updates them to fulfill with the constraints. The detail of the repair algorithm is described in Section 6.3. Finally, the list of solutions is collected and stored for further analysis.

8. Results

In this section, we present computational experiments with the proposed Spark binary cuckoo search algorithm. We test the algorithm on two classes of well-known problems.

- (1) OR-Library benchmarks: this class includes 65 small and medium size randomly generated problems that were frequently used in the literature. They are available in the OR-Library and are described in Table 1.
- (2) Railway scheduling problems: this class includes seven large-scale railway crew scheduling problems from Italian railways and are available in OR-Library.

Binary cuckoo search big data algorithm was implemented in python using Spark libraries. It was executed in Azure platform, Spark 1.6.1 and Hadoop 2.4.1 versions. To perform the statistical analysis in this study, the Wilcoxon signed-rank nonparametric test was used. For the results, each problem was executed 30 times.

The first stage corresponds to perform the parameter configuration used by the algorithm. To develop this activity,

TABLE 4: Average result by problem type for railway scheduling dataset.

Instance	Best known	BCSBA (5)	BCSBA (10)	BCSBA (20)	BCSBA (50)	BCSBA (100)	BCSBA (500)
Rail507	174	192.1	190.3	187.2	184.5	182.4	182.3
Rail516	182	189.4	187.2	183.2	183.2	182.4	182.4
Rail582	211	227.4	226.1	224.4	223.3	221.2	221.4
Rail2586	948	1152.3	1142.2	1140.1	1132.2	1130.8	1130.6
Rail2536	691	836.1	832.4	830.9	826.2	822.1	822.3
Average	441.2	519.46	515.64	513.16	509.88	507.58	507.6

TABLE 5: Average iteration by problem type.

Instance	BCSBA (5)	BCSBA (10)	BCSBA (20)	BCSBA (50)	BCSBA (100)	BCSBA (500)
4	82.2	80.1	80.6	76.3	74.9	76.9
5	81.7	76.4	76.4	77.1	70.6	71.4
6	88.6	87.8	86.4	84.7	82.5	82.3
A	112.5	109.1	108.2	106.3	97.8	98.8
B	105.6	107.5	100.2	101.1	99.2	96.6
C	130.5	128.3	120.1	110.8	108.4	105.4
D	134.7	132.1	130.9	125.4	109.9	107.6
E	138.1	131.6	130.5	124.5	109.5	112.4
F	145.6	136.4	135.2	115.6	108.4	107.6
G	297.3	276.4	271.6	254.1	196.5	194.6
H	267.1	256.6	245.1	221.2	160.7	161.8
Average	143.99	138.39	135.02	127.01	110.76	110.49

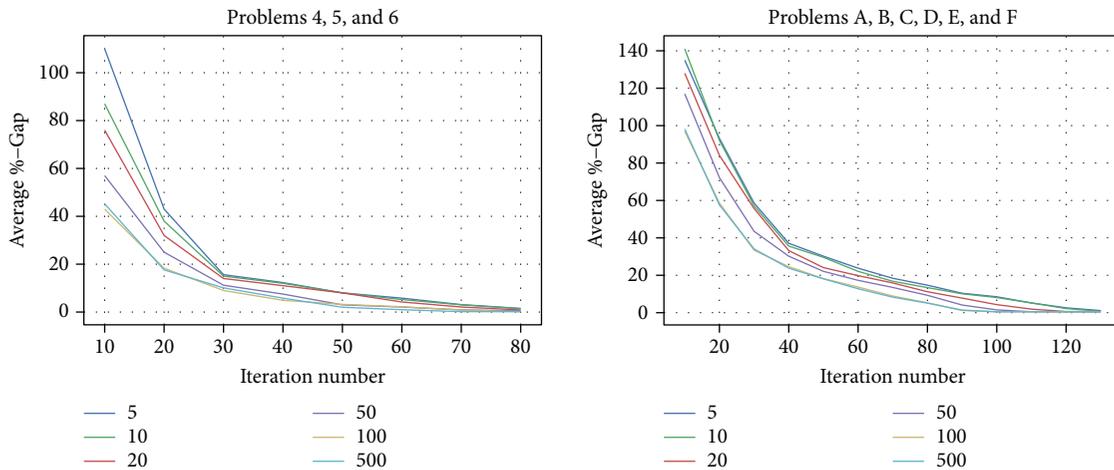


FIGURE 5: Convergence charts for instances of small and medium size problems.

the methodology described in [3] was used. In this methodology, four standard measures are used: the worst case, the best case, the average case, and the average execution time. With these four measurements, the area under the radar chart curve is obtained to define the best configuration. The dataset used to determine the best configuration corresponds to the first problem of each group $\{4.1, 5.1, \dots, G.1, H.1\}$. The results are shown in Table 2. In this table, the range column corresponds to the evaluated ranges and the value column to the value that will be used. The value of the parameters

γ , κ , and iteration number corresponds to those frequently used by the cuckoo search algorithm in the literature. The parameters α and K are specific to the K-means binarization method and are referenced to (10).

8.1. Evaluation of Result Quality through the Variation of the Solution Number. The goal of this section is to evaluate the number of solutions to be used by the binary cuckoo search big data algorithm (BCSBA) with respect to the quality of the results. For the execution of this experiment, the other

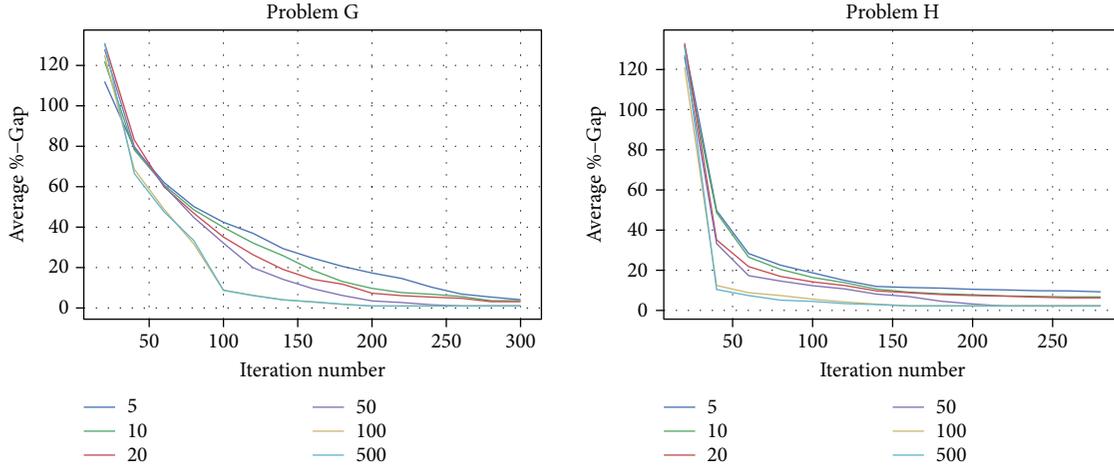


FIGURE 6: Convergence charts for instances of big size problems.

TABLE 6: Spark configuration.

Num-executors	Executor-cores	Executor-memory (Gb)
1	3	4
2	3	4
4	3	4
8	3	4
16	3	4

parameters used by the algorithm were the values described in the value column of the Table 2. In Table 3, the results are shown for cases that consider 5, 10, 20, 50, 100, and 500 solutions using the OR-Library dataset. From the table we observe that the results for cases 50, 100, and 500 are superior to the rest, nevertheless between them, they are very similar. Additionally, to see the significance, the Wilcoxon test was performed, comparing CSBA(5) with respect to other cases, obtaining that in all cases there is a significant difference. To complement the above analysis, violin charts were used to compare the distributions of the results through their shapes and interquartile ranges. The results are shown in Figure 4. The x -axis corresponds to the number of solutions used to solve the problem and the y -axis to %-Gap defined in (14). In the distributions, the superiority of the cases 50, 100, and 500 over the rest is appreciated. When we compare the cases 50, 100, and 500, between them, we see there is a similarity in the shape of their distributions as well as in the interquartile ranges.

$$\% \text{-Gap} = 100 \frac{\text{SolutionValue} - \text{BestKnown}}{\text{BestKnown}}. \quad (14)$$

In Table 4, the results for the railway scheduling problems are displayed. In this table, a behavior similar to the previous analysis is observed. The cases in which it uses 100 and 500 solutions obtained better results than the other cases. When comparing BCSBA-100 with BCSBA-500, similar results are observed.

8.2. Evaluation of Algorithm Convergence Time through the Solution Number. In this section, the convergence of the BCSBA algorithm with respect to the number of solutions is evaluated. For this analysis, the problems were grouped into 4 groups: the small group, which considers problems 4, 5, and 6; the median group, which considers problems A, B, C, D, E, and F; problem group G; and problem group H. Table 5 and Figures 5 and 6 show the results for different groups. In the table, it is observed that BCSBA has better convergence in cases 100 and 500 than in the rest of the cases, the result being very similar between 100 and 500. In Figures 5 and 6 the x -axis corresponds to the number of average iterations and the y -axis is the average of the %-Gap defined in (14). The data was collected every 10 iterations in the small and medium groups and every 20 iterations in the G and H groups. For the case of the small and medium groups, although the convergence curves are better in cases 100 and 500, the difference is quite small, which does not justify the increase in the number of solutions. For the case of the groups G and H, this difference becomes much more notorious.

8.3. Evaluation of Algorithm Scalability through Core Number. This last experiment aims to evaluate the scalability of our algorithm when considering more than one core for calculation. In Section 8.1 and Section 8.2, we see that increasing the number of solutions improves the results and decreases the number of iterations. However, the increase in the number of solutions has a computation cost. In this section, we evaluate whether the cost of computing can be diminished by the use of more processing cores.

In this section, we evaluate whether the cost of computing can be diminished by the use of more processing cores.

For the Spark configuration, three parameters were considered: num-executors which controls the number of executor requested, executor-cores property which controls the number of concurrent tasks an executor can run, and executor-memory which corresponds to the memory per executor. For the proper use of an executor, it is recommended to use between 3 and 5 cores. The considered Spark settings are shown in Table 6.

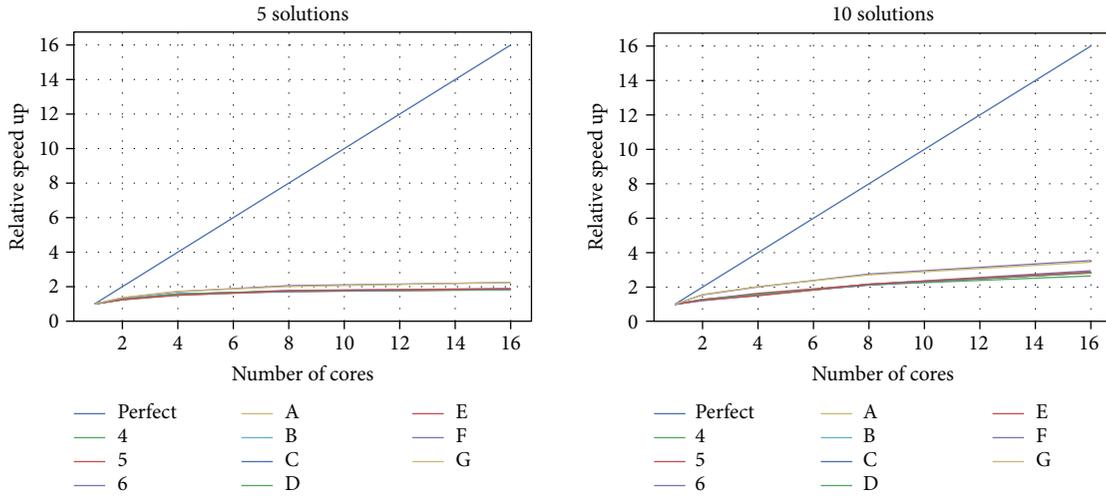


FIGURE 7: Speed up charts for 5 and 10 solutions.

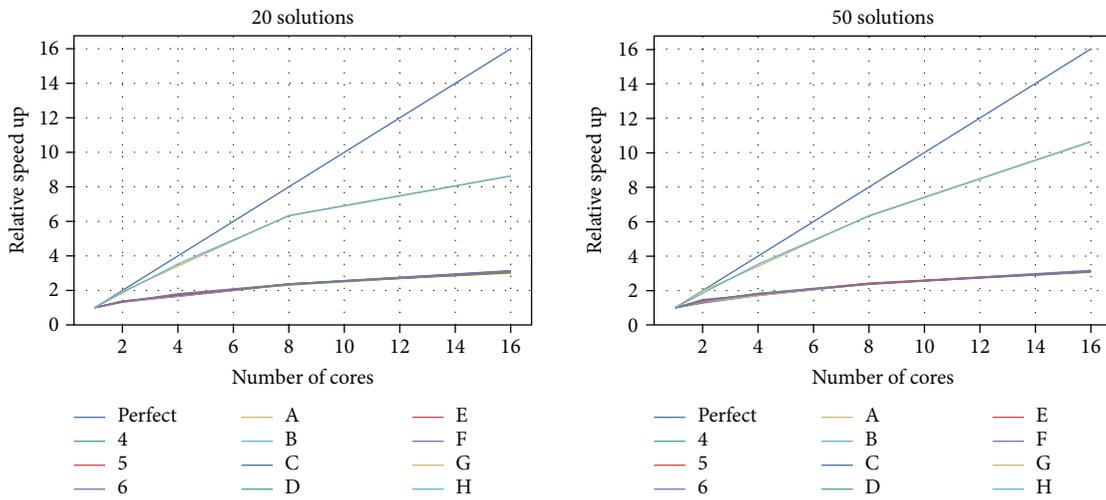


FIGURE 8: Speed up charts for 20 and 50 solutions.

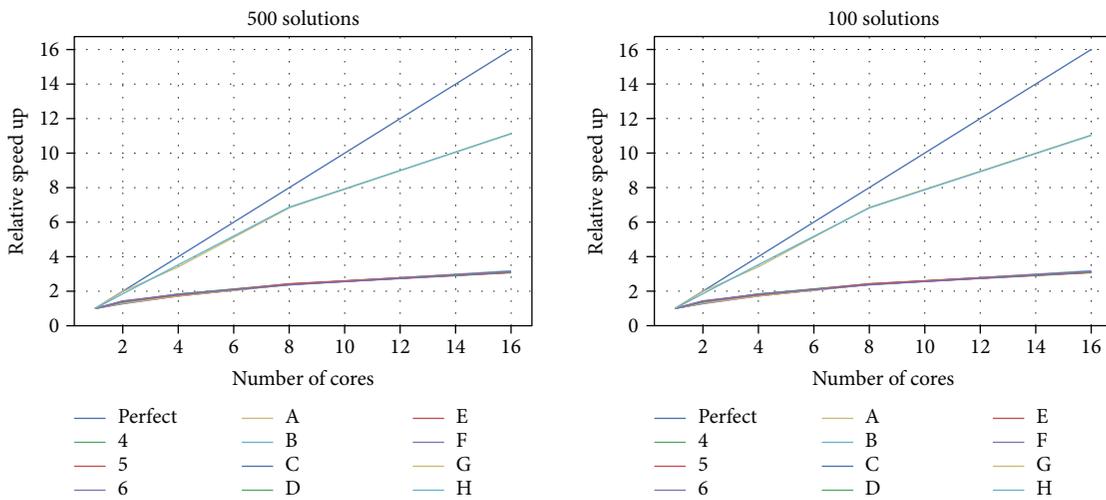


FIGURE 9: Speed up charts for 100 and 500 solutions.

In Figures 7, 8, and 9 we show the results of speed up charts for BCSBA using different numbers of solutions and considering between 1 and 16 executors. From the charts, it is observed that the best scalability is obtained for the case of 100 particles and in the problems G and H. For smaller problems, scalability is significantly reduced. The worst scalability was obtained for the algorithm using 5 particles. Another interesting fact is observed in the 500-particle chart where scalability was superior in G and H problems than in the rest; however, the performance is lower than in the case of 100 particles.

9. Conclusions

In this work, we have presented a binary cuckoo search big data algorithm applied to different instances of crew scheduling problem. We used an unsupervised learning method based on the K-means technique to perform binarization. Later, to develop the distributed version of the algorithm, Apache Spark was used as framework. The quality, convergence, and scalability of the results were evaluated in terms of the number of solutions used by the algorithm. It was found that quality, convergence, and scalability are affected by the number of solutions; however, these depend additionally on the problem that is being solved. In particular, it is observed that for medium size problems, the effects are not very relevant as opposed the large problems such as G and H, where the effect of the number of solutions is much more significant. On the other hand, when evaluating the scalability, we observe that it is also dependent on the number of solutions used by the algorithm and the size of the problems. The best performances were for problems G and H considering solutions between 20 and 500.

As a future work, it is interesting to investigate the proposed algorithm with other NP-hard problems with the intention of observing similar behaviours to observe in the case of CrSP. Also, we want to investigate how is the performance of autonomous search tuning algorithms [74] in big data environment. Finally, we also want to explore the performance of other metaheuristics in Big data Frameworks.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [2] J. Garca, B. Crawford, R. Soto, and G. Astorga, "A percentile transition ranking algorithm applied to binarization of continuous swarm intelligence metaheuristics," in *Recent Advances on Soft Computing and Data Mining. SCDM 2018. Advances in Intelligent Systems and Computing, Vol 700*, R. Ghazali, M. Deris, N. Nawi, and J. Abawajy, Eds., pp. 3–13, Springer, Cham, 2018.
- [3] J. García, B. Crawford, R. Soto, C. Castro, and F. Paredes, "A k-means binarization framework applied to multidimensional knapsack problem," *Applied Intelligence*, vol. 48, no. 2, pp. 357–380, 2018.
- [4] A. Franceschetti, E. Demir, D. Honhon, T. Van Woensel, G. Laporte, and M. Stobbe, "A metaheuristic for the time-dependent pollution-routing problem," *European Journal of Operational Research*, vol. 259, no. 3, pp. 972–991, 2017.
- [5] H. E. Nouri, O. B. Driss, and K. Ghédira, "Hybrid metaheuristics for scheduling of machines and transport robots in job shop environment," *Applied Intelligence*, vol. 45, no. 3, pp. 808–828, 2016.
- [6] R. Jovanovic, M. Tuba, and S. Voß, "A multi-heuristic approach for solving the pre-marshalling problem," *Central European Journal of Operations Research*, vol. 25, no. 1, pp. 1–28, 2017.
- [7] S. Basbug, "A general model for pattern synthesis of linear antenna arrays by metaheuristic algorithms," in *2017 International Applied Computational Electromagnetics Society Symposium - Italy (ACES)*, pp. 1–2, Florence, Italy, March 2017.
- [8] B. Crawford, R. Soto, E. Monfroy, G. Astorga, J. Garca, and E. Cortes, "A meta-optimization approach for covering problems in facility location," in *Applied Computer Sciences in Engineering. WEA 2017. Communications in Computer and Information Science*, J. Figueroa-García, E. López-Santana, J. Villa-Ramírez, and R. Ferro-Escobar, Eds., pp. 565–578, Springer, Cham, 2017.
- [9] J. Garca, B. Crawford, R. Soto, and P. Garca, "A multi dynamic binary black hole algorithm applied to set covering problem," in *Harmony Search Algorithm. ICHSA 2017. Advances in Intelligent Systems and Computing, Vol 514*, J. Ser, Ed., pp. 42–51, Springer, Singapore, 2017.
- [10] S. Santander-Jimenez and M. A. Vega-Rodriguez, "Asynchronous non-generational model to parallelize metaheuristics: a bioinformatics case study," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 1825–1838, 2017.
- [11] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets," in *Proceeding HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, p. 10, Boston, MA, USA, June 2010.
- [12] M. Zaharia, M. Chowdhury, T. Das et al., "Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing," in *Proceeding NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, p. 2, San Jose, CA, USA, April 2012.
- [13] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210–214, Coimbatore, India, December 2009.
- [14] A. K. Bhandari, V. K. Singh, A. Kumar, and G. K. Singh, "Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using kapur's entropy," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3538–3560, 2014.
- [15] J. Garca, B. Crawford, R. Soto, and G. Astorga, "A percentile transition ranking algorithm applied to knapsack problem," in *Applied Computational Intelligence and Mathematical Methods. CoMeSySo 2017. Advances in Intelligent Systems and Computing, Vol 662*, R. Silhavy, P. Silhavy, and Z. Prokopova, Eds., pp. 126–138, Springer, Cham, 2017.
- [16] S. M. Abd Elazim and E. S. Ali, "Optimal power system stabilizers design via cuckoo search algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 75, pp. 99–107, 2016.

- [17] S. Sudabattula and M. Kowsalya, "Optimal allocation of wind based distributed generators in distribution system using cuckoo search algorithm," *Procedia Computer Science*, vol. 92, pp. 298–304, 2016.
- [18] A. Kasirzadeh, M. Saddoune, and F. Soumis, "Airline crew scheduling: models, algorithms, and data sets," *EURO Journal on Transportation and Logistics*, vol. 6, no. 2, pp. 111–137, 2017.
- [19] K. Hoffmann, U. Buscher, J. S. Neufeld, and F. Tamke, "Solving practical railway crew scheduling problems with attendance rates," *Business & Information Systems Engineering*, vol. 59, no. 3, pp. 147–159, 2017.
- [20] S. Jütte, D. Müller, and U. W. Thonemann, "Optimizing railway crew schedules with fairness preferences," *Journal of Scheduling*, vol. 20, no. 1, pp. 43–55, 2017.
- [21] H. Öztop, U. Eliyi, D. T. Eliyi, and L. Kandiller, "A bus crew scheduling problem with eligibility constraints and time limitations," *Transportation Research Procedia*, vol. 22, pp. 222–231, 2017.
- [22] F. Quesnel, G. Desaulniers, and F. Soumis, "A new heuristic branching scheme for the crew pairing problem with base constraints," *Computers & Operations Research*, vol. 80, pp. 159–172, 2017.
- [23] O. Ö. Özener, M. Örmeci Matoğlu, G. Erdoğan, M. Haouari, and H. Sözer, "Solving a large-scale integrated fleet assignment and crew pairing problem," *Annals of Operations Research*, vol. 253, no. 1, pp. 477–500, 2017.
- [24] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.
- [25] E. Balas and M. C. Carrera, "A dynamic subgradient-based branch-and-bound procedure for set covering," *Operations Research*, vol. 44, no. 6, pp. 875–890, 1996.
- [26] J. E. Beasley, "An algorithm for set covering problem," *European Journal of Operational Research*, vol. 31, no. 1, pp. 85–93, 1987.
- [27] J. E. Beasley and P. C. Chu, "A genetic algorithm for the set covering problem," *European Journal of Operational Research*, vol. 94, no. 2, pp. 392–404, 1996.
- [28] J. E. Beasley, "A lagrangian heuristic for set-covering problems," *Naval Research Logistics*, vol. 37, no. 1, pp. 151–164, 1990.
- [29] A. Caprara, M. Fischetti, and P. Toth, "A heuristic method for the set covering problem," *Operations Research*, vol. 47, no. 5, pp. 730–743, 1999.
- [30] B. Yelbay, Ş. İlker Birbil, and K. Bülbül, "The set covering problem revisited: an empirical study of the value of dual information," *European Journal of Operational Research*, vol. 11, no. 2, pp. 575–594, 2015.
- [31] M. J. Brusco, L. W. Jacobs, and G. M. Thompson, "A morphing procedure to supplement a simulated annealing heuristic for cost-andcoverage-correlated set-covering problems," *Annals of Operations Research*, vol. 86, pp. 611–627, 1999.
- [32] C. Valenzuela, B. Crawford, R. Soto, E. Monfroy, and F. Paredes, "A 2-level metaheuristic for the set covering problem," *International Journal of Computers Communications & Control*, vol. 7, no. 2, pp. 377–387, 2014.
- [33] B. Crawford, R. Soto, N. Berríos et al., "A binary cat swarm optimization algorithm for the non-unicost set covering problem," *Mathematical Problems in Engineering*, vol. 2015, Article ID 578541, 8 pages, 2015.
- [34] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. tr 06, Erciyes university, engineering faculty, computer engineering department, 2005.
- [35] B. Crawford, R. Soto, G. Astorga, J. García, C. Castro, and F. Paredes, "Putting continuous metaheuristics to work in binary search spaces," *Complexity*, vol. 2017, Article ID 8404231, 19 pages, 2017.
- [36] R. C. Eberhart and J. Kennedy, "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pp. 4104–4108, Orlando, FL, USA, October 1997.
- [37] B. Crawford, M. Olivares-Suarez, W. Palma, F. Paredes, E. Olguin, and E. Norero, "A binary coded firefly algorithm that solves the set covering problem," *Romanian Journal of Information Science and Technology*, vol. 17, no. 3, pp. 252–264, 2014.
- [38] Y. Yang, Y. Mao, P. Yang, and Y. Jiang, "The unit commitment problem based on an improved firefly and particle swarm optimization hybrid algorithm," in *2013 Chinese Automation Congress*, pp. 718–722, Changsha, China, November 2013.
- [39] M. M. A. Aziz, T. M. Khalil, and H. K. M. Youseef, "A binary particle swarm optimization for optimal placement and sizing of capacitor banks in radial distribution feeders with distorted substation voltages," in *Proceedings of AIML International Conference*, pp. 137–143, Sharm El-Sheikh, Egypt, 2006.
- [40] D. G. Robinson, "Reliability analysis of bulk power systems using swarm intelligence," in *Annual Reliability and Maintainability Symposium, 2005. Proceedings*, pp. 96–102, Alexandria, VA, USA, January 2005.
- [41] J. Barraza, R. Soto, B. Crawford, R. Olivares, F. Johnson, and F. Paredes, "A binary cuckoo search algorithm for solving the set covering problem," in *Bioinspired Computation in Artificial Systems. IWINAC 2015. Lecture Notes in Computer Science, Vol 9108*, J. Ferrández Vicente, J. Álvarez-Sánchez, F. Paz López, F. Toledo-Moreo, and H. Adeli, Eds., pp. 88–97, Springer, Cham, 2015.
- [42] S. Palit, S. N. Sinha, M. A. Molla, A. Khanra, and M. Kule, "A cryptanalytic attack on the knapsack cryptosystem using binary firefly algorithm," in *2011 2nd International Conference on Computer and Communication Technology (ICCCCT-2011)*, pp. 428–432, Allahabad, India, September 2011.
- [43] K. Chandrasekaran and S. P. Simon, "Network and reliability constrained unit commitment problem using binary real coded firefly algorithm," *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 921–932, 2012.
- [44] X. Zhang, C. Wu, J. Li et al., "Binary artificial algae algorithm for multidimensional knapsack problems," *Applied Soft Computing*, vol. 43, pp. 583–595, 2016.
- [45] W. Liu, L. Li, and D. A. Cartes, "Angle modulated particle swarm optimization based defensive islanding of large scale power systems," in *2007 IEEE Power Engineering Society Conference and Exposition in Africa - PowerAfrica*, pp. 1–8, Johannesburg, South Africa, July 2007.
- [46] S. Das, R. Mukherjee, R. Kundu, and T. Vasilakos, "Multi-user detection in multi-carrier cdma wireless broadband system using a binary adaptive differential evolution algorithm," in *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference - GECCO '13*, pp. 1245–1252, Amsterdam, The Netherlands, July 2013.

- [47] Z. Dahi, C. Mezioud, A. Draa et al., *Computer Science and Its Applications. CIIA 2015. IFIP International Conference on Computer Science and Its Applications*, A. Amine, L. Bellatreche, Z. Elberrichi, E. Neuhold, and R. Wrembel, Eds., Springer, Cham, 2015.
- [48] G. Zhang, “Quantum-inspired evolutionary algorithms: a survey and empirical study,” *Journal of Heuristics*, vol. 17, no. 3, pp. 303–351, 2011.
- [49] Y. Zhou, I. Wang, and Y. Zhang, “Discrete quantum-behaved particle swarm optimization based on estimation of distribution for combinatorial optimization,” in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 897–904, Hong Kong, China, June 2008.
- [50] J. Zhao, J. Sun, and W. Xu, “A binary quantum-behaved particle swarm optimization algorithm with cooperative approach,” *International Journal of Computer Science*, vol. 10, no. 2, pp. 112–118, 2005.
- [51] J. Licheng, Y. Shuyuan, and W. Min, “A quantum particle swarm optimization,” in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, pp. 19–23, Portland, OR, USA, June 2004.
- [52] A. A. Ibrahim, A. Mohamed, H. Shareef, and S. P. Ghoshal, “An effective power quality monitor placement method utilizing quantum-inspired particle swarm optimization,” in *2011 International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 1–6, Bandung, Indonesia, July 2011.
- [53] A. R. Hota and A. Pat, “An adaptive quantum-inspired differential evolution algorithm for 0-1 knapsack problem,” in *2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pp. 703–708, Fukuoka, Japan, December 2010.
- [54] J. Alegria and Y. Túpac, “A generalized quantum-inspired evolutionary algorithm for combinatorial optimization problems,” in *XXXII International Conference of the Chilean Computer Science Society SCCC*, pp. 11–15, Temuco, Chile, November 2014.
- [55] S. Dey, S. Bhattacharyya, and U. Maulik, “New quantum inspired meta-heuristic techniques for multi-level colour image thresholding,” *Applied Soft Computing*, vol. 46, pp. 677–702, 2016.
- [56] A. Layeb, “A novel quantum inspired cuckoo search for knapsack problems,” *International Journal of Bio-Inspired Computation*, vol. 3, no. 5, pp. 297–305, 2011.
- [57] A. Layeb and S. R. Boussalia, “A novel quantum inspired cuckoo search algorithm for bin packing problem,” *International Journal of Information Technology and Computer Science*, vol. 4, no. 5, pp. 58–67, 2012.
- [58] A. Layeb, “A hybrid quantum inspired harmony search algorithm for 0-1 optimization problems,” *Journal of Computational and Applied Mathematics*, vol. 253, pp. 14–25, 2013.
- [59] Y. Zhou, X. Chen, and G. Zhou, “An improved monkey algorithm for a 0-1 knapsack problem,” *Applied Soft Computing*, vol. 38, pp. 817–830, 2016.
- [60] J. García, C. Pope, and F. Altimiras, “A distributed K -means segmentation algorithm applied to *Lobesia botrana* recognition,” *Complexity*, vol. 2017, Article ID 5137317, 14 pages, 2017.
- [61] E. Graells-Garrido and J. Garca, “Visual exploration of urban dynamics using mobile data,” in *Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information. UCAMi 2015. Lecture Notes in Computer Science, Vol 9454*, J. García-Chamizo, G. Fortino, and S. Ochoa, Eds., pp. 480–491, Springer, Cham, 2015.
- [62] E. Graells-Garrido, O. Peredo, and J. García, “Sensing urban patterns with antenna mappings: the case of Santiago, Chile,” *Sensors*, vol. 16, no. 7, p. 1098, 2016.
- [63] M. Minelli, M. Chambers, and A. Dhiraj, *Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for today's Businesses*, John Wiley & Sons, 2012.
- [64] O. F. Peredo, J. A. Garca, R. Stuvén, and J. M. Ortiz, “Urban dynamic estimation using mobile phone logs and locally varying anisotropy,” in *Geostatistics Valencia 2016. Quantitative Geology and Geostatistics, Vol 19*, J. Gómez-Hernández, J. Rodrigo-Ilarri, M. Rodrigo-Clavero, E. Cassiraga, and J. Vargas-Guzmán, Eds., pp. 949–964, Springer, Cham, 2017.
- [65] R. Shyam, H. B. Bharathi Ganesh, S. Sachin Kumar, P. Poornachandran, and K. P. Soman, “Apache Spark a big data analytics platform for smart grid,” *Procedia Technology*, vol. 21, pp. 171–178, 2015.
- [66] S. Sarraf and M. Ostadhashem, “Big data application in functional magnetic resonance imaging using apache spark,” in *2016 Future Technologies Conference (FTC)*, pp. 281–284, San Francisco, CA, USA, December 2016.
- [67] W. Zhou, R. Li, S. Yuan et al., “Metaspark: a spark-based distributed processing tool to recruit metagenomic reads to reference genomes,” *Bioinformatics*, vol. 33, no. 7, pp. 1090–1092, 2017.
- [68] Z. Zhang, K. Barbary, F. A. Nothhaft et al., “Kira: processing astronomy imagery using big data technology,” *IEEE Transactions on Big Data*, no. 1, p. 1, 2016.
- [69] L. Hao, S. Jiang, B. Si, and B. Bai, “Design of the research platform for medical information analysis and data mining,” in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1985–1989, Datong, China, October 2016.
- [70] Z. Guo, G. Fox, and M. Zhou, “Investigation of data locality in MapReduce,” in *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 419–426, Ottawa, ON, Canada, May 2012.
- [71] K. Grolinger, M. Hayes, W. A. Higashino, A. L’Heureux, D. S. Allison, and M. A. M. Capretz, “Challenges for MapReduce in Big Data,” in *2014 IEEE World Congress on Services*, pp. 182–189, Anchorage, AK, USA, June–July 2014.
- [72] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analysis*, O’Reilly Media, Inc., 2015.
- [73] D. Agrawal, S. Das, and A. El Abbadi, “Big data and cloud computing: current state and future opportunities,” in *Proceedings of the 14th International Conference on Extending Database Technology - EDBT/ICDT '11*, pp. 530–533, Uppsala, Sweden, March 2011.
- [74] Y. Hamadi, E. Monfroy, and F. Saubion, “What is autonomous search?,” in *Hybrid Optimization. Springer Optimization and Its Applications, Vol 45*, P. Hentenryck and M. Milano, Eds., pp. 357–391, Springer, New York, NY, USA, 2011.

Research Article

A Nonlinear Integer Programming Model for Integrated Location, Inventory, and Routing Decisions in a Closed-Loop Supply Chain

Hao Guo ¹, Congdong Li ¹, Ying Zhang ², Chunnan Zhang³, and Yu Wang⁴

¹School of Management, Jinan University, Guangzhou, Guangdong 510632, China

²Department of Industrial and Systems Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA

³Department of Accounting, Temple University, Philadelphia, PA 19122, USA

⁴International Business School, Jinan University, Zhuhai, Guangdong 519070, China

Correspondence should be addressed to Congdong Li; licd@jnu.edu.cn and Ying Zhang; yz29@buffalo.edu

Received 21 October 2017; Revised 29 May 2018; Accepted 10 June 2018; Published 22 July 2018

Academic Editor: Marek Reformat

Copyright © 2018 Hao Guo 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.

Facility location, inventory management, and vehicle routing are three important decisions in supply chain management, and location-inventory-routing problems consider them jointly to improve the performance and efficiency of today's supply chain networks. In this paper, we study a location-inventory-routing problem to minimize the total cost in a closed-loop supply chain that has forward and reverse logistics flows. First, we formulate this problem as a nonlinear integer programming model to optimize facility location, inventory control, and vehicle routing decisions simultaneously in such a system. Second, we develop a novel heuristic approach that incorporates simulated annealing into adaptive genetic algorithm to solve the model efficiently. Last, numerical analysis is presented to validate our solution approach, and it also provides meaningful managerial insight into how to improve the closed-loop supply chain under study.

1. Introduction

Supply chain management is critical for many business organizations to gain advantage in a competitive environment, and its impact has increased steadily in the past decades [1]. Although most practices in supply chain management were focused on forward logistics in early days, reverse logistics flows that are caused by consumer returns have gained a lot of attention recently and hence are considered by many firms to improve their business. According to a National Retail Federation report, the total merchandise returns accounted for \$260.5 billion and \$28.3 billion for the loss of the U.S. retailers and Canadian retailers in 2015, respectively [2]. Consumer returns also have a significant impact on e-commerce, and it is shown that at least 30% of all the products ordered online are returned as compared to 8.89% in traditional offline stores [3]. Particularly, for fashion products such as fashion apparel, the return rate can be as high as 75% [4]. Therefore, consumer returns

represent a growing financial and operational concern for many firms in different industries, and they also have a significant impact on their supply chains.

Closed-loop supply chains (CLSCs) [5, 6] are an emerging topic in supply chain management because of the growing concern about consumer returns and environmental sustainability. Unlike traditional supply chains that only consider forward logistics flows directed from manufacturers to consumers, CLSCs also consist of reverse flows of new or used products that are directed from consumers to manufacturers. In practice, business managers need to make many strategic, tactical, and operational decisions such as facility locations, inventory control, and vehicle routing decisions to improve the efficiency and sustainability of their supply chains. In this paper, we study a location-inventory-routing problem (LIRP) that integrates those three decisions in a multiechelon closed-loop supply chain network for a manufacturer. This network comprises a manufacturing factory, multiple hybrid distribution-collection centers

(HDCCs), and several retailers, where HDCCs will operate as warehouses and collection centers in the forward and reverse flows, respectively. From a practical perspective, the research questions that motivate this study are summarized as follows:

- (1) For a manufacturer, how to decide HDCC locations in a supply chain network when forward and reverse logistics flows are both considered, and how to use those HDCCs to fulfill the demands and collect returns from retailers?
- (2) What is the optimal stock replenishment policy for those HDCCs?
- (3) How to optimize vehicle routes in the forward and reverse flows when retailers are served by those HDCCs?

The rest of this paper is organized as follows: Section 2 reviews the related literature. Section 3 describes the research problem under study and formulates it as a nonlinear integer programming model. Section 4 proposes an adaptive hybrid simulated annealing genetic algorithm (AHSAGA) to solve the model efficiently. Section 4.1 presents the numerical study and computational results. Section 5 concludes this paper and provides directions for future research.

2. Literature Review

The location-inventory-routing problem (LIRP) comprises three subproblems: facility location, inventory control, and vehicle routing. Since they are highly correlated in the real-world business, many research efforts have been conducted to study those problems jointly. The location inventory problem (LIP) is the integrated form of the first two problems, and they are first proposed by Daskin et al. [7] and Shen et al. [8]. LIPs have been extended with many business scenarios such as lateral transshipment [9], perishable products [10], correlated demands [11], disruption risk [12], and inventory control strategies [13], and most of those works are reviewed by Farahani et al. [14]. Recently, LIPs are also studied by incorporating CLSCs. For example, Diabat et al. [15] study a LIP by considering spare parts in a closed-loop system, Guo et al. [16] study location-inventory decisions for closed-loop supply chain management with secondary market consideration, and Li et al. [17] present an important and meaningful work by studying LIP and CLSC with third-party logistics (3PL) because it is a fundamental logistics strategy that has been adopted by many firms in practice. The location routing problems (LRPs) integrate facility location and vehicle routing problems but ignore inventory management decisions. Min et al. [18] and Nagy and Salhi [19] review the research works related to LRPs in early days, and Schneider and Drexel [20] examine the most recent works that are published in the literature since the survey by Nagy and Salhi [19].

LIRPs incorporate all three decisions above, and hence they are a more comprehensive form. In early days, Shen and Qi [21] develop a location-allocation model which

approximates routing costs according to the locations of opened depots, and then Javid and Azad [22] study such a problem without any approximation. Moreover, LIRPs are extensively studied under many practical settings such as perishable products [23, 24], deterministic or stochastic demand [25–27], and disruption risks [28]. Closed-loop supply chains (CLSCs) have attracted considerable attention from researchers and practitioners because of the significant impact of consumer returns, and it is emergent to study LIRPs under a CLSC setting. For example, Li et al. [29] study a LIRP by considering returns in an electronic supply chain environment, and Deng et al. [30] develop and solve a model when returned products can be either defective or nondefective. From the perspective of sustainability, Zhalechian et al. [31] design a closed-loop system with location routing inventory decisions under mixed uncertainty.

In this paper, a nonlinear integer program model is formulated to study a LIRP in a closed-loop logistics system by considering many real-world business scenarios such as vehicle capacity and the disposal of different types of returned products. To solve this model efficiently, we develop a novel solution approach that extends the power of the adaptive genetic algorithm by incorporating simulated annealing, and numerical study shows that it is more powerful and efficient than other similar heuristics in the literature.

3. The Model

3.1. Problem Description. In this paper, we study a closed-loop supply chain network that comprises a manufacturing factory, multiple hybrid distribution-collection centers (HDCCs), and several retailers. This network can be represented by a directed graph in which vertices are the factory, HDCCs, and retailers, and the edges can be directed from the factory to retailers via HDCCs, or vice versa. More specifically, in the forward flow, new products are first shipped from the factory to HDCCs and then from HDCCs to retailers by vehicles on certain routes. In the reverse flow, returned products are sent from retailers to HDCCs for inspection first. A returned product will be disposed immediately at a HDCC if it cannot be refurbished. Otherwise, it will be sent from HDCCs to the factory for repair. In this system, HDCCs operate as warehouses and return collection centers on working days in the forward and reverse flows, respectively. Vehicles are used to deliver new products from HDCCs to retailers as well as to collect returned products from retailers to HDCCs, and a vehicle must return to the same HDCC after it visits all retailers on a route. Figure 1 illustrates the closed-loop supply chain network under study.

For simplicity, we consider a single type of products and vehicles and assume that a retailer will be assigned to a same HDCC in the forward and reverse flows. Given the locations of the factory and retailers, HDCCs will be built at selected locations, and a HDCC will order new products from the factory and serve at least one retailer in the forward and reverse flows. To minimize the total cost in this system, the following decisions will be optimized:

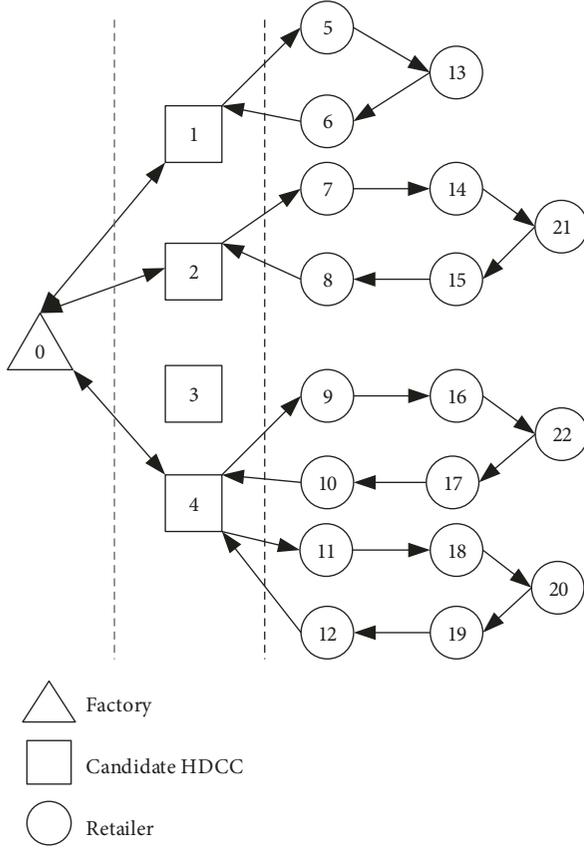


FIGURE 1: A closed-loop supply chain network.

- (1) HDCC location and retailer assignment: selecting locations to build HDCCs and assigning retailers to those HDCCs
- (2) Inventory replenishment: deciding the optimal order frequency and quantity for each HDCC
- (3) Vehicle routing: designing circular vehicle routes starting from and ended by each HDCC

3.2. Objective Function. In the closed-loop supply chain under study, the total cost is composed by the following: (1) location cost which is the fixed cost of building and operating HDCCs; (2) working inventory cost including order, holding, and shipping costs; (3) routing cost between HDCCs and retailers; (4) return cost. The individual costs per year are calculated as follows:

- (1) Location cost: $C_{LOC} = \sum_{r \in R} a_r W_r$
- (2) Working inventory cost

The working inventory cost comprises three individual terms. The first term is the order cost that is incurred when placing orders to the factory at HDCCs, the second term is the holding cost of new products in inventory, and the third term is the shipping cost of new products from the factory to

HDCCs. Similar to [22], we adopt a (Q, r) inventory model with type I service to manage inventories at HDCCs, and the holding cost is adapted from the standard form in the economic order quantity (EOQ) model. Obviously, the order frequency and quantity at a HDCC is determined by the expected demands of the retailers that are served by the HDCC. Therefore, the individual terms of the working inventory cost can be written as follows:

- (i) Order cost: $\sum_{r \in R} f_r N_r$
- (ii) Holding cost of new products: $\sum_{r \in R} \sum_{i \in S} (h \lambda d_i X_{ir} / 2N_r)$
- (iii) Shipping cost from the factory to HDCCs: $\sum_{r \in R} e_r N_r + \sum_{r \in R} \sum_{i \in S} b_r \lambda d_i X_{ir}$

Consequently, the total working inventory cost per year is given as follows:

$$C_{INV} = \sum_{r \in R} (f_r + e_r) N_r + \lambda \sum_{r \in R} \sum_{i \in S} \left(\frac{h d_i}{2N_r} + b_r d_i \right) X_{ir}. \quad (1)$$

(3) Vehicle routing cost

- (i) Forward logistics: $\sum_{r \in R} \sum_{v \in V} \sum_{i \in S} u \lambda d_i s_{ri} Y_{irv}$
- (ii) Reverse logistics: $\sum_{r \in R} \sum_{v \in V} \sum_{i \in S} u \lambda q_i t_{ir} Y_{irv}$

Therefore, the total annual routing cost is given as follows:

$$C_{VRT} = \lambda u \sum_{r \in R} \sum_{v \in V} \sum_{i \in S} (s_{ri} d_i + t_{ir} q_i) Y_{irv}. \quad (2)$$

(4) Return cost

- (i) Inspection cost: $\sum_{r \in R} \sum_{i \in S} p_r \lambda q_i X_{ir}$
- (ii) Disposal cost at HDCCs: $\sum_{r \in R} \sum_{i \in S} g_r \theta \lambda q_i X_{ir}$
- (iii) Cost of refurbish returned products at the factory: $\sum_{i \in S} m(1 - \theta) \lambda q_i$
- (iv) Shipment cost from HDCCs to the factory: $\sum_{r \in R} \sum_{i \in S} b_r \lambda (1 - \theta) q_i X_{ir}$
- (v) Holding cost of returned products: $\sum_{r \in R} \sum_{i \in S} k \lambda q_i X_{ir}$

For simplicity, we assume that the holding cost of a returned product is independent of how long it stays in inventory. Therefore, the total return cost per year is given as follows:

$$C_{RET} = \sum_{i \in S} m(1 - \theta) \lambda q_i + \lambda \sum_{r \in R} \sum_{i \in S} (p_r + g_r \theta + b_r (1 - \theta)) q_i X_{ir}. \quad (3)$$

According to the individual costs above, the total annual cost in the CLSC is calculated as follows:

$$\begin{aligned}
C = & \sum_{r \in R} a_r W_r + \lambda u \sum_{r \in R} \sum_{v \in V} \sum_{i \in S} (s_{ri} d_i + t_{ir} q_i) Y_{irv} + \sum_{r \in R} (f_r + e_r) N_r \\
& + \lambda \sum_{r \in R} \sum_{i \in S} \left(\frac{h d_i}{2 N_r} + b_r d_i X_{ir} \right) + \sum_{i \in S} m(1 - \theta) \lambda q_i \\
& + \lambda \sum_{r \in R} \sum_{i \in S} (p_r + g_r \theta + b_r(1 - \theta) + k) q_i X_{ir}.
\end{aligned} \tag{4}$$

Therefore, the location-inventory-routing problem under study can be formulated as follows:

$$\begin{aligned}
\min \quad & C, \\
\text{subject to} \quad & \sum_{r \in R} W_r \geq 1, \tag{5}
\end{aligned}$$

$$\sum_{r \in R} U_{rv} \leq 1 \quad \forall v \in V, \tag{6}$$

$$\sum_{r \in R} X_{ir} = 1 \quad \forall i \in S, \tag{7}$$

$$\sum_{r \in R} \sum_{v \in V} Y_{irv} = 1 \quad \forall i \in S, \tag{8}$$

$$\sum_{r \in R} \sum_{i \in S} d_i Y_{irv} \leq c \quad \forall v \in V, \tag{9}$$

$$U_{rv} \leq W_r \quad \forall r \in R, \forall v \in V, \tag{10}$$

$$\sum_{v \in V} U_{rv} \geq W_r \quad \forall r \in R, \tag{11}$$

$$X_{ir} \leq W_r \quad \forall i \in S, \forall r \in R, \tag{12}$$

$$\sum_{i \in S} X_{ir} \geq W_r \quad \forall r \in R, \tag{13}$$

$$\sum_{v \in V} U_{rv} \leq \sum_{i \in S} X_{ir} \quad \forall r \in R, \tag{14}$$

$$Y_{irv} \leq U_{rv} \quad \forall i \in S, \forall r \in R, \forall v \in V, \tag{15}$$

$$\sum_{i \in S} Y_{irv} \geq U_{rv} \quad \forall r \in R, \forall v \in V, \tag{16}$$

$$Y_{irv} \leq X_{ir} \quad \forall i \in S, \forall r \in R, \forall v \in V, \tag{17}$$

$$\sum_{v \in V} Y_{irv} = X_{ir} \quad \forall i \in S, \forall r \in R, \tag{18}$$

$$\sum_{j \in L} Z_{ijrv} = Y_{irv} \quad \forall i \in S, \forall r \in R, \forall v \in V, \tag{19}$$

$$\sum_{i \in L} Z_{ijrv} = Y_{jrv} \quad \forall j \in S, \forall r \in R, \forall v \in V, \tag{20}$$

$$\sum_{j \in S} Z_{rjrv} \leq \sum_{i \in S} Y_{irv} \quad \forall r \in R, \forall v \in V, \tag{21}$$

$$\sum_{j \in S} Z_{jrv} \leq \sum_{i \in S} Y_{irv} \quad \forall r \in R, \forall v \in V, \tag{22}$$

$$M \sum_{j \in S} Z_{rjrv} \geq \sum_{i \in S} Y_{irv} \quad \forall r \in R, \forall v \in V, M \text{ is a big number}, \tag{23}$$

$$M \sum_{j \in S} Z_{jrv} \geq \sum_{i \in S} Y_{irv}, \quad \forall r \in R, \forall v \in V, M \text{ is a big number}, \tag{24}$$

$$\sum_{j \in S} Z_{rjrv} \leq 1 \quad \forall r \in R, \forall v \in V, \tag{25}$$

$$\sum_{j \in S} Z_{jrv} \leq 1 \quad \forall r \in R, \forall v \in V, \tag{26}$$

$$\sum_{i \in L} Z_{ikrv} - \sum_{j \in L} Z_{kjrv} = 0 \quad \forall k \in L, \forall r \in R, \forall v \in V, \tag{27}$$

$$Z_{ijrv} = 0 \quad \forall i \in S, \forall j, r \in R, j \neq r, \forall v \in V, \tag{28}$$

$$Z_{ijrv} + Z_{jrv} = 1 \quad \forall i, j \in S, \forall r \in R, \forall v \in V, \tag{29}$$

$$W_r \in \{0, 1\} \quad \forall r \in R, \tag{30}$$

$$U_{ir} \in \{0, 1\} \quad \forall r \in R, \forall v \in V, \tag{31}$$

$$X_{ir} \in \{0, 1\} \quad \forall i \in S, \forall r \in R, \tag{32}$$

$$Y_{irv} \in \{0, 1\} \quad \forall i \in S, \forall r \in R, \forall v \in V, \tag{33}$$

$$Z_{ijrv} \in \{0, 1\} \quad \forall i \in W, \forall j \in W, \forall r \in R, \forall v \in V. \tag{34}$$

The constraints of this model are explained as follows. Constraint (5) means that at least one HDCC will be built. Constraint (6) means that a vehicle can be assigned to at most one HDCC. Constraint (7) means that a retailer will be served by exactly one HDCC. Constraint (8) means that a retailer will be placed in exactly one route. Constraint (9) is the vehicle capacity constraint which means that the total demand in a route cannot exceed the capacity of a vehicle. Constraint (10) means that vehicles can be assigned to a HDCC only if the HDCC has been built. Constraint (11) means that at least one vehicle will be assigned to a HDCC. Constraint (12) means that retailers can be served by a HDCC only if it has been built. Constraint (13) means that at least one retailer will be served by a HDCC. Constraint (14) means that for each HDCC, the number of vehicles or routes is less than the number of retailers. Constraints (15) and (16) enforce the relationship between U_{rv} and Y_{irv} , and they mean that a retailer can be placed in a route only if the route exists as well as that at least one retailer will be included in a route. Constraints (17) and (18) enforce the relationship between X_{ir} and Y_{irv} , and they mean that a retailer will be placed in exactly one route which belongs to a HDCC if and only if it is served by the same HDCC. Constraints (19) and (20) enforce the relationship between Y_{irv} and Z_{ijrv} for retailers, and they mean that a retailer cannot have neighbor locations in a route if it is not in the route as well as that no closed subloop will present in a route. Constraints (21), (22), (23), and (24) enforce the relationship between Y_{irv} and Z_{ijrv} for HDCCs, and they mean that exactly one HDCC

will be placed in each route. Constraints (25) and (26) mean that in a route, a HDCC can be directed to/from at most one retailer. Constraint (27) is the flow conservation constraint, and it means that a vehicle must leave a retailer after it arrives at this retailer and hence the route is circular. Constraint (28) means that a route cannot be directed from a retailer to a HDCC if the retailer is not served by this HDCC. Constraint (29) guarantees that a route will be in one direction but not in two directions. Constraints (30), (31), (32), (33), and (34) specify that W_r , U_{rv} , X_{ir} , Y_{irv} , and Z_{ijrv} are binary variables.

It is obvious that the objective function is convex with respect to N_r . To calculate the optimal number of orders placed at HDCC r annually, let $\partial C/\partial N_r = 0$, then we have

$$N_r^* = \sqrt{\frac{\sum_{v \in V} \sum_{i \in S} \lambda h d_i Y_{irv}}{2(e_r + f_r)}}. \quad (35)$$

By substituting N_r^* into (4), the objective function can be rewritten as

$$\begin{aligned} C' = & \sum_{r \in R} a_r W_r + \lambda u \sum_{r \in R} \sum_{v \in V} \sum_{i \in S} (s_{ri} d_i + t_{ir} q_i) Y_{irv} \\ & + \sqrt{2\lambda h \sum_{r \in R} \sum_{i \in S} (f_r + e_r) d_i X_{ir}} + \lambda \sum_{r \in R} \sum_{i \in S} b_r d_i X_{ir} \\ & + \sum_{i \in S} m(1 - \theta) \lambda q_i + \lambda \sum_{r \in R} \sum_{i \in S} (p_r + g_r \theta + b_r(1 - \theta) + k) q_i X_{ir}. \end{aligned} \quad (36)$$

4. Solution Approach

Facility location and vehicle routing problems are NP-hard in general [22], and LIRPs can be more complex to solve because of the integration of those problems. In this paper, we propose a two-phase heuristic method that incorporates simulated annealing (SA) into adaptive genetic algorithm (AGA). More specifically, facility location and vehicle routing decisions will be encoded as chromosomes in AGA, and then the two decisions will be optimized by an evolution process. Thereafter, the optimal inventory replenishment decision will be determined accordingly.

GA is a popular search technique to solve optimization problems based on the principles of natural selection and genetics [32]. In practice, a GA process may converge prematurely or do not converge at all, both of which will lead to bad solutions, and hence adaptive coefficients are usually used to compensate those shortcomings. SA is a probabilistic method which was first proposed to find the global minimum of a cost function that may possess several local minima [33], and it has been widely used to solve many research problems. In this paper, we propose a novel heuristic algorithm, that is, adaptive hybrid simulated annealing genetic algorithm (AHSAGA), to solve the nonlinear integer programming model presented in Section 3. AHSAGA is an improved form of traditional AGAs by adopting the great local search capacity of SA, and our numerical experiments show that it is an effective approach in terms of both solution accuracy and time efficiency.

4.1. Basis

4.1.1. Encoding and Decoding. When GA is applied to solve an optimization problem, chromosomes are usually used to represent the candidate solutions to this problem, and they will evolve to better solutions iteratively. In this study, the solutions to the location and routing problems will be first encoded as chromosomes and then solved by AHSAGA. Once the location and routing problems are solved, inventory decisions can be easily optimized by solving (35).

The length of a chromosome is $R + S$, where R and S are the number of candidate HDCC locations and retailers, respectively. Let N be the population size, then an initial population can be created by randomly choosing N chromosomes that satisfy (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19), (20), (21), (22), (23), (24), (25), (26), (27), (28), (29), (30), (31), (32), (33), and (34). In a chromosome, a HDCC and its subsequent retailers comprise a circular route. Therefore, each chromosome will start with a candidate HDCC location. If a chromosome starts with a retailer, then the first allele will be replaced by a candidate HDCC location because the retailers before the first candidate HDCC location in the chromosome will not be assigned to any HDCC. Moreover, if there are consecutive candidate HDCC locations in a chromosome, then a HDCC will be built at the location represented by the last allele of this string.

4.1.2. Fitness Function and Selection Method. Once a population is created, chromosomes or candidate solutions will be evaluated by their fitness to decide whether they will be kept in its offspring population. In this study, the fitness of an individual is measured as follows:

$$f_k = \frac{1}{C'}, \quad 1 \leq k \leq M, \quad (37)$$

where C' is the objective function given by (36).

In AHSAGA, roulette-wheel selection is adopted to select and copy solutions with higher fitness values into new populations. Let N be the population size, then chromosome i will be reproduced in the next generation if it satisfies the equation below:

$$\frac{\sum_{k=1}^{i-1} f_k}{\sum_{k=1}^N f_k} < \xi_i \leq \frac{\sum_{k=1}^i f_k}{\sum_{k=1}^N f_k}, \quad (38)$$

where f_k is the fitness value of chromosome k , $\xi_i \in [0, 1]$ is a random number that follows the uniform distribution.

4.1.3. Crossover Operator. In general, a GA process will start with an initial population that is generated randomly, and the fitness of solutions will be improved iteratively by applying selection, crossover, mutation, and replacement operators. In AHSAGA, crossover operator will be applied in an iteration by the following three steps to recombine individuals for a better offspring:

- (1) Choose two parents from a population randomly and decide two crossover points arbitrarily.

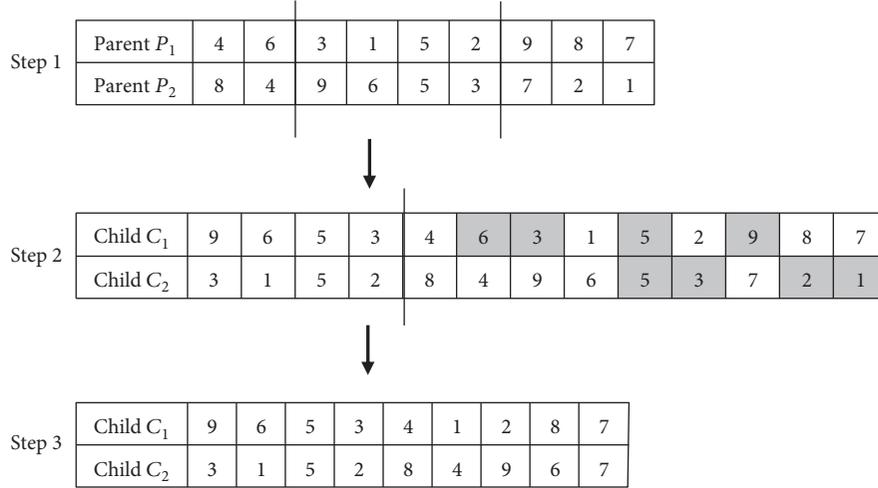


FIGURE 2: An example of crossover operation.

- (2) Generate two intermediate chromosomes by moving all the alleles positioned between the crossover lines in a parent to the beginning of the other.
- (3) In each intermediate chromosome, remove the same alleles which appear in the string moved from the other parent.

An example of this procedure is illustrated in Figure 2.

Usually, fitness values of chromosomes will be significantly different at the beginning of a GA process, and hence crossover is greatly beneficial to speed up the evolution. In AHSAGA, the probability of crossover is given by (39), which is similar to [34] in spirit.

$$p_c = p_{c0} + \alpha_c \frac{(f_{\text{avg}})^{n_c}}{(f_{\text{max}} - f_{\text{min}})^{n_c} + (f_{\text{avg}})^{n_c}}, \quad (39)$$

where p_{c0} is the initial probability of crossover, α_c and n_c are the two adaptive coefficients, f_{max} , f_{avg} , and f_{min} are the maximal, average, and minimal fitness values in a population, respectively.

4.1.4. Mutation Operator. Crossover operators cannot work effectively if individuals have similar fitness values in a population. For example, in some cases, new chromosomes cannot be generated by crossover if two parents have the same allele at a given gene. To solve this problem, mutation is designed to add diversity to the population and make it possible to explore the entire search space [32]. AHSAGA uses an inverse function as the mutation operator to select two points in a parent chromosome randomly and invert the order of the alleles between the two points. For example, if (7 8 5 | 3 6 9 1 | 4 10 2) is a parent, then the first and second split points are located after the third and seventh genes, respectively, and hence the offspring will be (7 8 5 | 1 9 6 3 | 4 10 2) after inverse. Mutation will

occur randomly, and the probability of mutation is given as follows:

$$p_m = p_{m0} + \alpha_m \frac{(f_{\text{max}} - f_{\text{min}})^{n_m}}{(f_{\text{max}} - f_{\text{min}})^{n_m} + (f_{\text{avg}})^{n_m}}, \quad (40)$$

where p_{m0} is the initial probability of mutation, α_m and n_m are the two adaptive coefficients, f_{max} , f_{avg} , and f_{min} are the maximal, average, and minimal fitness values in a population, respectively.

If a chromosome starts with a retailer, then the initial allele will be inverted with the first allele that represents a candidate HDCC location.

4.1.5. Simulated Annealing and Individual Replacement. In AGAs, individuals will be replaced by new ones for evolution. AHSAGA adopts SA as the steady-state technique [32], and the probability that a chromosome will be replaced is given as follows:

$$p = \begin{cases} 1, & \text{if } f_{\text{new}} \geq f_{\text{old}}, \\ \exp\left(-\frac{(1/f_{\text{new}})/(1/f_{\text{old}})}{T}\right), & \text{if } f_{\text{new}} < f_{\text{old}}, \end{cases} \quad (41)$$

where f_{new} and f_{old} are the fitness values of the new and old individuals, respectively, and T is the temperature given as follows:

$$T_{t+1} = \alpha T_t, \quad t \geq 0, 0 < \alpha < 1, \quad (42)$$

where T_t is the temperature at time t , and α is the change rate of temperature.

4.2. Algorithm. The pseudocode of AHSAGA is shown in Algorithm 1, and the steps in this algorithm are briefly explained as follows:

Step 1. Initialize parameters such as population size pop_size , iteration number M , crossover factors p_{c0} , α_c , n_c , mutation

```

Input: Parameters in Section 4.1
Output: Optimal location-route decisions
Begin
  Choose population size  $pop\_size$ ;
  Create an initial population  $pop(0)$  randomly;
  for ( $i = 1$  to  $pop\_size$ ) {
    Calculate fitness  $f_i$  for individual  $i$  in population  $pop(0)$ ;
  }
   $f_{\max}(0) = \max \{f_i\}$ ;
   $f_{\min}(0) = \min \{f_i\}$ ;
   $f_{avg}(0) = \text{sum}(f_i)/pop\_size$ ;
   $f_{best} = f_{\max}(0)$ ;
   $pop_{best} = \text{individual with } f_{best}$ ;
  Choose  $M$  ( $M > 0$ ) as the number of iterations;
   $m = 1$ ;
  while ( $m \leq M$ ) {
     $n = 1$ ;
    while ( $n \leq pop\_size$ ) {
      Apply select operator to create the mating pool;
      Choose two chromosomes (parents) from the mating pool randomly;
      Generate random number  $r_c$ ;
      Calculate crossover probability  $p_c$ ;
      if ( $r_c \leq p_c$ ) {
        Apply crossover operator;
      }
      Generate random number  $r_m$ ;
      Calculate mutation probability  $p_m$ ;
      if ( $r_m \leq p_m$ ) {
        Apply mutation operator;
      }
    }
    for ( $j = 1$  to  $pop\_size$ ) {
      Calculate fitness  $f_j$  for individual  $j$  in population  $pop(m)$ ;
    }
     $f_{\max}(m) = \max \{f_j\}$ ;
     $f_{\min}(m) = \min \{f_j\}$ ;
    if ( $f_{\max}(m) \geq f_{\max}(m-1)$ ) then {
       $f_{best} = f_{\max}(m)$ ;
       $pop_{best} = \text{individual with } f_{best}$ ;
    }
    else {
       $\Delta = 1/f_{\max}(m) - 1/f_{\max}(m-1)$ ;
       $z = e^{-\Delta/T}$ ;
      Generate random number  $r_z$ ;
      If ( $z \leq r_z$ ) then {
         $f_{best} = f_{\max}(m)$ ;
         $pop_{best} = \text{individual with } f_{best}$ ;
      }
      else {
        Replace the individual with  $f_{\min}(m)$  by  $pop_{best}$  in population  $pop(m)$ ;
      }
       $T = \alpha T$ ;
    }
     $m = m + 1$ ;
  }
end

```

ALGORITHM 1: AHSAGA.

factors p_{m0} , α_m , n_m , initial temperature T_0 , and cooling factor α .

Step 2. Create an initial population randomly.

Step 3. Generate an offspring population by applying selection, crossover, and mutation operators.

Step 4. Calculate the fitness values of the individuals in a new population, and identify those with the maximal and minimal fitness values.

Step 5. Check whether the maximal fitness value in an offspring population is greater than that in its parent population. If yes, go to Step 6. Otherwise, apply SA to decide whether the best solution in the parent population will be introduced into the offspring population, then update temperature in SA.

Step 6. Check whether the termination condition is satisfied. If yes, return the chromosome with the maximal fitness value. Otherwise, go to Step 3.

5. Numerical Study

In this study, AHSAGA is implemented by Matlab R2014a and all numerical experiments are conducted on a workstation equipped with an Intel Core i7-4790 CPU at 3.60 GHz and 8.0 GB of RAM under Windows 7.

To validate its performance, AHSAGA has been tested on five data sets that are adapted from LRP files provided by the University of Aveiro [35] for locations, fixed costs, and demands. For example, the input data adapted from the Gaskell67-21 \times 5 files for HDCCs and retailers are shown in Tables 1 and 2, respectively. All other parameters are provided in Table 3.

5.1. Sensitivity Analysis. Since the performance of AHSAGA can be affected significantly by its parameters, a sensitivity analysis is conducted on the parameters shown in Table 4 to identify the optimal setting in this study. To eliminate the excessive number of combinations, other parameters will be set to their median values when a parameter is tested. The numerical results are shown in Figure 3, where red and blue lines represent the mean objective values and average computational times, respectively. Figure 3 shows that solution accuracy and computational times are both affected by those parameters, and the best result can be archived when $M = 1000$, $N = 100$, $p_{c0} = 0.8$, $p_{m0} = 0.25$, $\alpha_c = 0.09$, $\alpha_m = 0.175$, $n_c = 3$, $n_m = 2$, $T_0 = 200$, and $\alpha = 0.98$. The optimal setting is presented under the ‘‘Experiment setting’’ column in Table 4, and it will be used in the subsequent experiments.

From Figure 3, we can see that AHSAGA can be affected by those parameters in the following way:

- (1) M and N

When M or N increases, the optimal value and computational time will decrease and increase,

TABLE 1: Gaskell67-21 \times 5 (retailer).

Retailer	Coordinates	Demand
i_1	(151,264)	55
i_2	(159,261)	35
i_3	(130,254)	40
i_4	(128,252)	70
i_5	(163,247)	105
i_6	(146,246)	20
i_7	(161,242)	40
i_8	(142,239)	5
i_9	(163,236)	25
i_{10}	(148,232)	30
i_{11}	(128,231)	60
i_{12}	(156,217)	65
i_{13}	(129,214)	65
i_{14}	(146,208)	15
i_{15}	(164,208)	45
i_{16}	(141,206)	105
i_{17}	(147,193)	50
i_{18}	(164,193)	45
i_{19}	(129,189)	125
i_{20}	(155,185)	90
i_{21}	(139,182)	35

Remark: in this table, the daily demands are adapted by dividing the original quantities by 20 due to the vehicle capacity parameter used in this study.

TABLE 2: Gaskell67-21 \times 5 (HDCC).

Depot	Coordinates	Fixed cost
1	(136,194)	50
2	(143,237)	50
3	(136,216)	50
4	(137,204)	50
5	(128,197)	50

respectively. This indicates that more iterations or greater population diversity will lead to a better optimal solution with an additional time cost.

- (2) p_{c0} and p_{m0}

The optimal value will always decrease when p_{m0} increases, but it will not always decrease when p_{c0} increases. This indicates that a larger probability of mutation is always helpful to get a better optimal solution, but the probability of crossover should be moderately large. Moreover, we can see that when p_{c0} and p_{m0} increase, the computational time will increase and decrease, respectively. This indicates that a larger p_{c0} and p_{m0} will slow down and speed up the convergence of AHSAGA, respectively.

- (3) a_c and a_m

TABLE 3: LIRP parameters.

Parameter	Description	Value
b_r	Shipping cost per unit of product between a manufacturing plant and HDCC r	$U [6, 10]$
c	Vehicle capacity	1500
e_r	Fixed cost per shipment from a plant to HDCC r	$U [21, 25]$
f_r	Fixed administrative and handling cost of placing an order to a plant at HDCC r	$U [16, 20]$
g_r	Disposal cost per unit of returned product which cannot be refurbished at HDCC r	2
h	Holding cost per unit of new product per year at HDCC r	2
k	Holding cost per unit of returned product at HDCC r	1
m	Fixed cost of repairing and repacking one unit of returned product at a manufacturing plant	2
p_r	Inspection cost per unit of returned product at HDCC r	1
q_i	Daily returns from retailer i	$U [1, 5]$
u	Shipping cost per unit of product and distance	5
θ	Probability that a returned product cannot be refurbished	0.3
λ	Working days per year	300

TABLE 4: AHSAGA parameters.

Parameter	Description	Sensitivity analysis		Experimental setting
		Range	Median	
M	Number of iterations	{200, 400, 600, 800, 1000}	600	600
N	Population size	{20, 40, 60, 80, 100}	60	60
p_{c0}	Initial crossover probability	{0.5, 0.6, 0.7, 0.8, 0.9}	0.7	0.8
p_{m0}	Initial mutation probability	{0.05, 0.1, 0.15, 0.2, 0.25}	0.15	0.25
α_c	Adaptive coefficient	{0.08, 0.09, 0.1, 0.11, 0.12}	0.1	0.09
α_m	Adaptive coefficient	{0.1, 0.125, 0.15, 0.175, 0.2}	0.15	0.175
n_c	Adaptive coefficient	{1, 2, 3, 4, 5}	3	3
n_m	Adaptive coefficient	{1, 2, 3, 4, 5}	3	2
T_0	Initial temperature	{50, 100, 150, 200, 250}	150	200
α	Cooling rate	{0.95, 0.96, 0.97, 0.98, 0.99}	0.97	0.98

When a_c increases, the optimal solution can be always improved with an additional time cost. But a_m should have a moderate value to achieve the best performance in terms of optimal solution and computational time.

(4) n_c and n_m

n_c needs to be moderate to get the best optimal solution, but a larger n_c can always improve the convergence and reduce the computational time. When n_m increases, the optimal solution can always be improved with an additional time cost.

(5) T_0 and α

A higher initial temperature T_0 in SA can always reduce computational time, but it is not always helpful to get a better optimal solution. However, a larger cooling factor α will always improve the optimal solution and computational time.

5.2. Illustrative Example. In this section, Gaskell67-21 \times 5 files [35] are used as an example to show the application and performance of AHSAGA. To get started, an initial chromosome is generated randomly, which is {5, 9, 10, 8, 26, 7, 14, 21, 15, 12, 13, 25, 1, 2, 11, 16, 3, 4, 18, 20, 6, 17, 24, 22, 19, 23}. According to the encoding-decoding scheme in Section 4.1, HDCC locations and vehicle routes can be decoded as that in Table 5, and then the corresponding optimal number of orders per year can be calculated by (36). When the algorithm is executed iteratively, objective values and adaptive probabilities change monotonically as shown in Figures 4 and 5, respectively.

To validate its performance, AHSAGA is compared with other two heuristics in the literature, which are adaptive annealing genetic algorithm (IAGA) [34] and hybrid genetic simulated annealing algorithm (HGSAA) [29]. To avoid any bias, each algorithm is replicated 50 times by using a same data set, and the mean objective values and computational times are compared. To illustrate the stochastic nature of

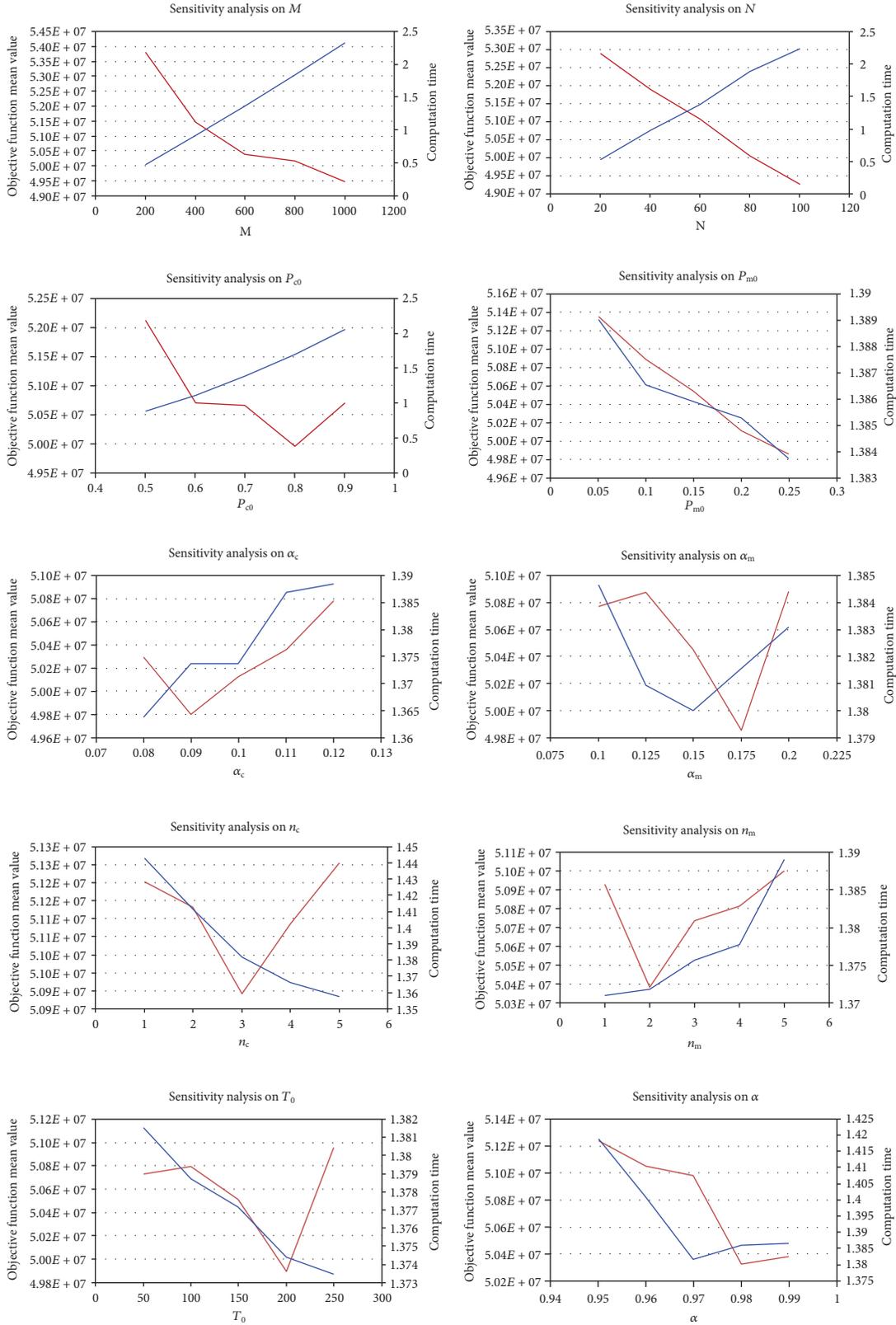


FIGURE 3: Sensitivity analysis on AHSAGA parameters.

the three algorithms, Figure 6 shows the 50 objective values from AHSAGA, HGSAA, and IAGA in a descending order by using the data adapted from Gaskell67-21 \times 5

files, and Table 6 presents a more thorough comparison between the three algorithms on this data set, which shows that AHSAGA is more effective than HGSAA and IAGA

TABLE 5: Initial HDCC locations and vehicle routes.

HDCC number	Vehicle number	Route	Number of orders
2	1	11-16	24
4	2	18-20-6-17	62
	3	24-22-19-23	
5	4	9-10-8-26-7-14	65
	5	21-15-12-13-25	

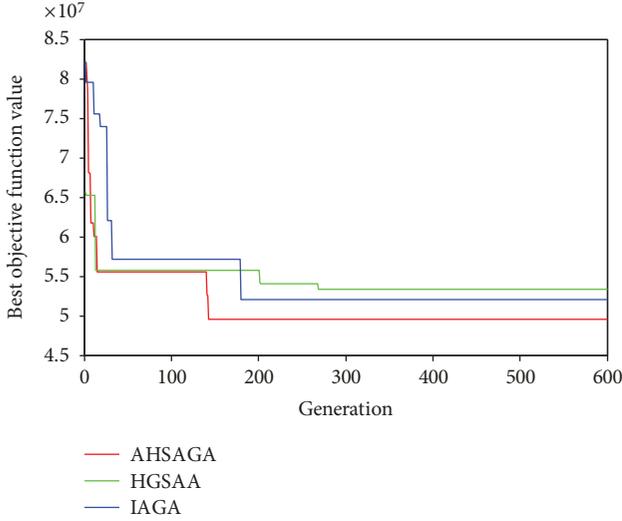
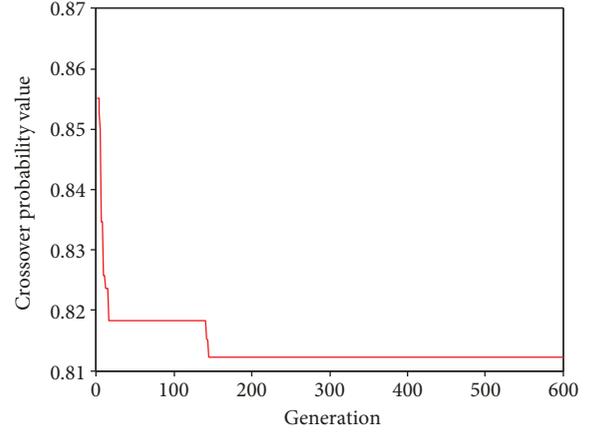


FIGURE 4: Trend in objective values.

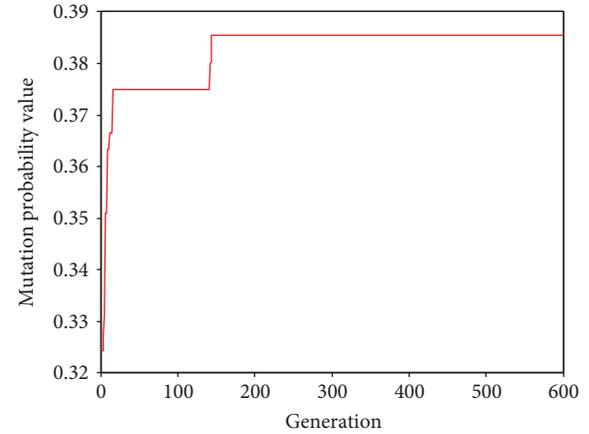
from the perspectives of robustness, solution quality, and time efficiency.

5.3. *Performance Comparison.* The section presents a comprehensive comparison between AHSAGA, HGSAA, and IAGA on three types of problems by the number of retailers. More specifically, the number of retailers is less than 50 in small-size problems, between 50 and 100 in medium-size problems, and more than 100 in large-size problems. The numerical results on small-size, medium-size, and large-size problems are shown in Tables 7–9, respectively, from which we can make the following conclusions:

- (i) The mean objective values from AHSAGA are significantly lower than those from IAGA and HGSAA for most problems. This indicates that AHSAGA has a great capability to search global optimums and hence can provide better solutions.
- (ii) AHSAGA takes less computational times and convergence generations to find the optimal solution than IAGA and HGSAA for all problems. This indicates that AHSAGA is the most efficient approach.
- (iii) The variation of the optimal values from AHSAGA, which is measured by the coefficient of variation, is



(a) Crossover probability



(b) Mutation probability

FIGURE 5: Trend in adaptive probabilities.

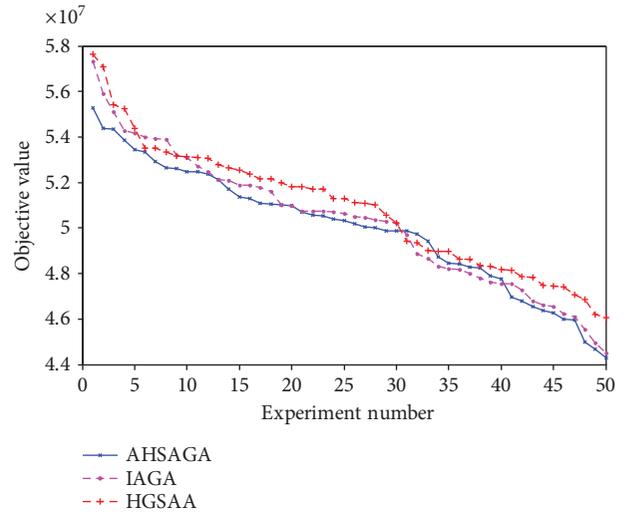


FIGURE 6: Objective values from AHSAGA, HGSAA, and IAGA (Gaskell67-21 \times 5).

lower than that from IAGA and HGSAA for all problems. This indicates that AHSAGA is more robust and consistent than the other two algorithms.

TABLE 6: Computational results on Gaskell67-21 \times 5.

Instance name	HGSAA			IAGA			AHSAGA				
	M1	SD	CV	M2	SD	CV	M3	SD	CV	$\frac{(M1-M3)}{M1}$	$\frac{(M2-M3)}{M2}$
Computational time	2.09	0.02	0.01	1.85	0.03	0.02	1.76	0.02	0.01	15.59%	4.80%
Convergence generation	221.22	121.62	0.55	211.02	112.90	0.54	202.18	107.53	0.53	8.61%	4.19%
Total cost	50860938.99	2778602.09	0.05	50276017.05	2991285.98	0.06	49977998.24	2709926.51	0.05	1.74%	0.59%

Remark: in Table 6, M1, M2, and M3 represent the mean objective values obtained by HGSAA, IAGA, and AHSAGA, respectively. SD means standard deviation, and CV means coefficient of variation. The same convention will also be used in the subsequent tables.

TABLE 7: Computational results for small-size problems.

Instance name	HGSAA			IAGA			AHSAGA			$\frac{(M1 - M3)}{M1}$	$\frac{(M2 - M3)}{M2}$	
	M1	SD	CV	M2	SD	CV	M3	SD	CV			
Srivastava8-8 × 2	Computational time	1.30	0.02	1.09	0.01	0.01	1.10	0.01	0.01	0.01	15.73%	-0.43%
	Convergence generation	201.66	116.19	203.34	115.63	115.63	195.14	108.22	0.55	0.55	3.23%	4.03%
	Total cost	58357282.29	3318088.97	0.06	58972563.90	3408120.16	0.06	58080565.56	3150230.79	0.05	0.05	0.47%
Perl83-12 × 2	Computational time	1.39	0.02	1.20	0.02	0.01	1.19	0.01	0.01	0.01	14.24%	1.01%
	Convergence generation	197.06	109.60	199.30	133.01	133.01	187.28	116.90	0.62	0.62	4.96%	6.03%
	Total cost	9879584.53	514520.39	0.05	9866462.28	494570.55	0.05	9857315.61	490727.85	0.05	0.05	0.23%
Gaskell67-22 × 5	Computational time	2.01	0.04	1.71	0.03	0.02	1.71	0.03	0.01	0.01	14.99%	0.11%
	Convergence generation	206.52	115.21	217.36	123.96	123.96	198.64	109.48	0.55	0.55	3.82%	8.61%
	Total cost	90392692.15	4533650.78	0.05	90140435.61	5993545.10	0.07	89430743.81	4280679.66	0.05	0.05	1.06%
Min92-27 × 5	Computational time	2.37	0.01	1.99	0.02	0.01	1.99	0.02	0.01	0.01	16.38%	0.00%
	Convergence generation	175.82	106.58	188.10	103.53	103.53	175.68	94.84	0.54	0.54	0.08%	6.60%
	Total cost	363157783.99	25693304.31	0.07	367201239.25	27220166.35	0.07	356654161.93	24405194.09	0.07	0.07	1.79%
Gaskell67-29 × 5	Computational time	2.51	0.03	2.08	0.03	0.01	2.07	0.02	0.01	0.01	17.74%	0.87%
	Convergence generation	212.40	117.98	208.40	134.58	134.58	200.86	101.74	0.51	0.51	5.43%	3.62%
	Total cost	85966292.78	4976794.74	0.06	85879276.51	5292913.64	0.06	85053221.84	4861829.37	0.06	0.06	1.06%
Gaskell67-32 × 5	Computational time	2.62	0.04	2.28	0.03	0.01	2.24	0.03	0.01	0.01	14.48%	1.55%
	Convergence generation	202.96	119.95	201.96	119.24	119.24	197.00	113.53	0.58	0.58	2.94%	2.46%
	Total cost	142731259.91	7358411.37	0.05	142691917.33	7239503.55	0.05	140407259.50	7051863.18	0.05	0.05	1.63%
Gaskell67-36 × 5	Computational time	2.86	0.04	2.45	0.04	0.02	2.45	0.03	0.01	0.01	14.33%	0.03%
	Convergence generation	192.58	113.85	205.52	122.26	122.26	189.00	111.51	0.59	0.59	1.86%	8.04%
	Total cost	51830112.18	3370888.67	0.07	52045789.31	2998029.49	0.06	51240375.41	2951876.85	0.06	0.06	1.14%

TABLE 8: Computational results for medium-size problems.

Instance name	HGSAA			IAGA			AHSAGA				
	MI	SD	CV	M2	SD	CV	M3	SD	CV	$\frac{(M1 - M3)}{M1}$	$\frac{(M2 - M3)}{M2}$
Christofides69-50 × 5	Computational time	3.52	0.04	0.01	2.96	0.05	0.02	0.04	0.01	15.60%	-0.14%
	Convergence generation	185.52	118.83	0.64	179.58	114.49	0.64	106.46	0.61	6.26%	3.16%
	Total cost	64622673.96	4896193.44	0.08	65120367.91	4880581.60	0.07	63410536.90	4447087.25	0.07	1.88%
Perl83-55 × 15	Computational time	4.55	0.04	0.01	4.14	0.06	0.01	0.05	0.01	10.01%	1.27%
	Convergence generation	202.68	120.89	0.60	199.34	117.33	0.59	114.00	0.59	3.97%	2.36%
	Total cost	75608999.15	4218565.18	0.06	75488435.46	4821373.57	0.06	74318631.82	3372606.19	0.05	1.71%
Christofides69-75 × 10	Computational time	5.53	0.06	0.01	4.76	0.09	0.02\	0.05	0.01	13.24%	-0.73%
	Convergence generation	203.52	119.05	0.58	201.04	115.46	0.57	110.48	0.57	5.08%	3.91%
	Total cost	140932062.97	10696126.94	0.08	137838194.63	13260464.21	0.10	137205765.38	10092485.60	0.07	2.64%
Perl83-85 × 7	Computational time	6.03	0.04	0.01	5.21	0.11	0.02	0.10	0.02	13.48%	-0.11%
	Convergence generation	211.78	119.39	0.56	186.32	113.21	0.61	102.34	0.56	14.00%	2.24%
	Total cost	407102768.27	10738366.84	0.03	404838974.84	12150698.81	0.03	404614214.54	9236869.46	0.02	0.61%
Daskin95-88 × 8	Computational time	6.23	0.04	0.01	5.47	0.10	0.02	0.08	0.02	12.84%	0.75%
	Convergence generation	214.16	125.73	0.59	206.94	112.33	0.54	110.85	0.54	3.51%	0.14%
	Total cost	95326029.80	6322807.23	0.07	94077056.19	5903902.59	0.06	94002405.72	5837160.63	0.06	1.39%

TABLE 9: Computational results for large-size problems.

Instance name	HGSAA			IAGA			AHSAGA				
	M1	SD	CV	M2	SD	CV	M3	SD	CV	$\frac{(M1-M3)}{M1}$	$\frac{(M2-M3)}{M2}$
Christofides69-100 × 10	Computational time	7.19	0.08	0.01	6.55	0.14	0.02	0.07	0.01	9.97%	1.18%
	Convergence generation	190.72	114.09	0.60	191.74	116.64	0.61	106.69	0.57	1.31%	1.84%
	Total cost	229586535.05	13462701.82	0.06	231296925.10	15532578.99	0.07	227749748.70	10551026.83	0.05	0.80%
Or76-117 × 14	Computational time	8.70	0.07	0.01	7.71	0.20	0.03	0.10	0.01	11.34%	-0.08%
	Convergence generation	209.78	126.74	0.60	198.64	119.05	0.60	108.04	0.57	10.25%	5.22%
	Total cost	5062462547.35	255122955.85	0.05	5079364715.42	256027550.46	0.05	5010781647.96	244879865.55	0.05	1.02%
Min92-134 × 8	Computational time	9.57	0.10	0.01	8.53	0.17	0.02	0.10	0.01	11.40%	0.61%
	Convergence generation	191.82	133.67	0.70	188.68	111.54	0.59	106.78	0.58	3.91%	2.31%
	Total cost	3727217265.35	200966165.84	0.05	3750482276.62	201259019.17	0.05	3721752794.82	166680491.26	0.04	0.15%
Daskin95-150 × 10	Computational time	10.95	0.10	0.01	9.07	0.19	0.02	0.13	0.01	17.22%	0.05%
	Convergence generation	223.24	112.92	0.51	204.74	107.05	0.52	95.69	0.49	12.55%	4.65%
	Total cost	17531706417.34	1394570344.68	0.08	14896666385.68	1140033562.24	0.08	14688680807.96	1050208955.68	0.07	16.22%
Per183-318 × 4	Computational time	9.50	0.10	0.01	8.40	0.18	0.02	0.08	0.01	11.64%	0.03%
	Convergence generation	195.98	113.32	0.58	189.50	105.69	0.56	102.74	0.55	4.39%	1.12%
	Total cost	3758638540.59	174594029.53	0.05	3764348485.23	174118187.75	0.05	3718507764.20	166085079.53	0.04	1.07%

6. Conclusions and Future Study

Closed-loop supply chains are an emerging and important topic due to the tremendous economic and environmental impact of consumer returns. In this paper, we study a location-inventory-routing problem in a closed-loop supply chain by formulating it as a nonlinear integer programming model. Since the problem is NP-hard, we also design a novel adaptive genetic algorithm by incorporating simulated annealing to solve this model efficiently. To make this study more practical, many real-world business scenarios such as vehicle capacity and the disposal of different types of returned products are also considered and modeled precisely.

This study can be extended in several directions in the future: first, this problem will be more practical and flexible if some assumptions are relaxed. For example, it will be flexible to allow a many-to-many relationship between vehicles and retailers, and it will also be more practical to relax the assumption that a retailer will be visited by a vehicle every working day. Second, since secondary markets have become an important channel to sell used products, it will be greatly beneficial to study LIRPs in a CLSC by considering those markets. Third, our model will be more valuable if it incorporates more business scenarios such as supply risk and multi-sourcing.

Sets

- R : set of candidate HDCC locations, where $r \in R$
- V : set of vehicles, where $v \in V$
- S : set of retailers, where $i, j \in S$
- L : set of locations, which is the union of HDCCs and retailers (i.e., $L = R \cup S$).

Parameters

- a_r : fixed cost of building and operating a HDCC at location r
- b_r : shipping cost per unit of product between the factory and HDCC r
- c : vehicle capacity
- d_i : daily demand of retailer i
- e_r : fixed cost per shipment from the factory to HDCC r
- f_r : fixed administrative and handling cost of placing an order to the factory from HDCC r
- g_r : disposal cost per unit of returned product which cannot be refurbished at HDCC r
- h : holding cost per unit of new product per year at HDCC r
- k : holding cost per unit of returned product at HDCC r
- m : fixed cost of repairing and repacking one unit of returned product at the factory
- p_r : inspection cost per unit of returned product at HDCC r
- q_i : daily returns from retailer i , where $q_i < d_i$
- s_{ri} : distance from HDCC r to retailer i in a route (forward logistics)
- t_{ir} : distance from retailer i to HDCC r in a route (reverse logistics)
- u : shipping cost per unit of product and distance
- θ : probability that a returned product cannot be refurbished

λ : workdays per year (remark: similar to [21], we assume that a retailer will be visited by a vehicle every workday. Hence, λ is also the number of road trips for a vehicle per year).

Decision Variables

N_r : number of orders placed at HDCC r per year

$$W_r = \begin{cases} 1, & \text{if a HDCC is built at location } r, \\ 0, & \text{otherwise.} \end{cases}$$

$$U_{rv} = \begin{cases} 1, & \text{if vehicle } v \text{ is operated by the} \\ & \text{HDCC at location } r, \\ 0, & \text{otherwise.} \end{cases}$$

$$X_{ir} = \begin{cases} 1, & \text{if retailer } i \text{ is served by the} \\ & \text{HDCC at location } r, \\ 0, & \text{otherwise.} \end{cases}$$

$$Y_{irv} = \begin{cases} 1, & \text{if the logistics flows between} \\ & \text{retailer } i \text{ and the HDCC at location} \\ & r \text{ are carried by vehicle } v, \\ 0, & \text{otherwise.} \end{cases}$$

$$Z_{ijrv} = \begin{cases} 1, & \text{if vehicle } v \text{ is directed from} \\ & \text{retailer } i \text{ to } j \text{ on a route that belongs} \\ & \text{to the HDCC at location } r, \\ 0, & \text{otherwise.} \end{cases}$$

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant nos. 71672074 and 71772075.

References

- [1] T. P. Harrison, "Principles for the strategic design of supply chains," in *The practice of supply chain management: Where theory and application converge*, pp. 3–12, Springer, Boston, MA, USA, 2004.
- [2] National Retail Federation, "2015 consumer returns in the retail industry," April 2018, https://nrf.com/sites/default/files/Images/Media%20Center/NRF%20Retail%20Return%20Fraud%20Final_0.pdf.
- [3] "E-commerce product return rate - statistics and trends," May 2018, <https://www.invespcro.com/blog/ecommerce-product-return-rate-statistics/>.
- [4] J. Mostard and R. Teunter, "The newsboy problem with resalable returns: a single period model and case study," *European Journal of Operational Research*, vol. 169, no. 1, pp. 81–96, 2006.

- [5] M. Al-Salem, A. Diabat, D. Dalalah, and M. Alrefaei, "A closed-loop supply chain management problem: reformulation and piecewise linearization," *Journal of Manufacturing Systems*, vol. 40, pp. 1–8, 2016.
- [6] Y. C. Tsao, V. T. Linh, and J. C. Lu, "Closed-loop supply chain network designs considering RFID adoption," *Computers & Industrial Engineering*, vol. 113, pp. 716–726, 2017.
- [7] M. S. Daskin, C. R. Coullard, and Z. J. M. Shen, "An inventory-location model: formulation, solution algorithm and computational results," *Annals of Operations Research*, vol. 110, no. 1/4, pp. 83–106, 2002.
- [8] Z.-J. M. Shen, C. Coullard, and M. S. Daskin, "A joint location-inventory model," *Transportation Science*, vol. 37, no. 1, pp. 40–55, 2003.
- [9] J. Meissner and O. V. Senicheva, "Approximate dynamic programming for lateral transshipment problems in multi-location inventory systems," *European Journal of Operational Research*, vol. 265, no. 1, pp. 49–64, 2018.
- [10] Z. Dai, F. Aqlan, X. Zheng, and K. Gao, "A location-inventory supply chain network model using two heuristic algorithms for perishable products with fuzzy constraints," *Computers & Industrial Engineering*, vol. 119, pp. 338–352, 2018.
- [11] B. Vahdani, M. Soltani, M. Yazdani, and S. Meysam Mousavi, "A three level joint location-inventory problem with correlated demand, shortages and periodic review system: robust meta-heuristics," *Computers & Industrial Engineering*, vol. 109, pp. 113–129, 2017.
- [12] M. Farahani, H. Shavandi, and D. Rahmani, "A location-inventory model considering a strategy to mitigate disruption risk in supply chain by substitutable products," *Computers & Industrial Engineering*, vol. 108, pp. 213–224, 2017.
- [13] S. J. Sadjadi, A. Makui, E. Dehghani, and M. Pourmohammad, "Applying queuing approach for a stochastic location-inventory problem with two different mean inventory considerations," *Applied Mathematical Modelling*, vol. 40, no. 1, pp. 578–596, 2016.
- [14] R. Z. Farahani, H. Rashidi Bajgan, B. Fahimnia, and M. Kaviani, "Location-inventory problem in supply chains: a modelling review," *International Journal of Production Research*, vol. 53, no. 12, pp. 3769–3788, 2015.
- [15] A. Diabat, T. Abdallah, and A. Henschel, "A closed-loop location-inventory problem with spare parts consideration," *Computers & Operations Research*, vol. 54, pp. 245–256, 2015.
- [16] H. Guo, C. Li, Y. Zhang, C. Zhang, and M. Lu, "A location-inventory problem in a closed-loop supply chain with secondary market consideration," *Sustainability*, vol. 10, no. 6, p. 1891, 2018.
- [17] Y. Li, H. Guo, and Y. Zhang, "An integrated location-inventory problem in a closed-loop supply chain with third-party logistics," *International Journal of Production Research*, vol. 56, no. 10, pp. 3462–3481, 2018.
- [18] H. Min, V. Jayaraman, and R. Srivastava, "Combined location-routing problems: a synthesis and future research directions," *European Journal of Operational Research*, vol. 108, no. 1, pp. 1–15, 1998.
- [19] G. Nagy and S. Salhi, "Location-routing: issues, models and methods," *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.
- [20] M. Schneider and M. Drexel, "A survey of the standard location-routing problem," *Annals of Operations Research*, vol. 259, no. 1-2, pp. 389–414, 2017.
- [21] Z.-J. Max Shen and L. Qi, "Incorporating inventory and routing costs in strategic location models," *European Journal of Operational Research*, vol. 179, no. 2, pp. 372–389, 2007.
- [22] A. Ahmadi Javid and N. Azad, "Incorporating location, routing and inventory decisions in supply chain network design," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 5, pp. 582–597, 2010.
- [23] A. Hiassat, A. Diabat, and I. Rahwan, "A genetic algorithm approach for location-inventory-routing problem with perishable products," *Journal of Manufacturing Systems*, vol. 42, pp. 93–103, 2017.
- [24] Z. Rafie-Majd, S. H. R. Pasandideh, and B. Naderi, "Modeling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm," *Computers & Chemical Engineering*, vol. 109, pp. 9–22, 2018.
- [25] W. J. Guerrero, C. Prodhon, N. Velasco, and C. A. Amaya, "Hybrid heuristic for the inventory location-routing problem with deterministic demand," *International Journal of Production Economics*, vol. 146, no. 1, pp. 359–370, 2013.
- [26] Y. Zhang, M. Qi, L. Miao, and E. Liu, "Hybrid metaheuristic solutions to inventory location routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, pp. 305–323, 2014.
- [27] N. Nekooghadirli, R. Tavakkoli-Moghaddam, V. R. Ghezavati, and S. Javanmard, "Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics," *Computers & Industrial Engineering*, vol. 76, pp. 204–221, 2014.
- [28] F. Rayat, M. M. Musavi, and A. Bozorgi-Amiri, "Bi-objective reliable location-inventory-routing problem with partial back-ordering under disruption risks: a modified amosa approach," *Applied Soft Computing*, vol. 59, pp. 622–643, 2017.
- [29] Y. Li, H. Guo, L. Wang, and J. Fu, "A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under e-supply chain environment," *The Scientific World Journal*, vol. 2013, Article ID 125893, 10 pages, 2013.
- [30] S. Deng, Y. Li, H. Guo, and B. Liu, "Solving a closed-loop location-inventory-routing problem with mixed quality defects returns in e-commerce by hybrid ant colony optimization algorithm," *Discrete Dynamics in Nature and Society*, vol. 2016, Article ID 6467812, 12 pages, 2016.
- [31] M. Zhalechian, R. Tavakkoli-Moghaddam, B. Zahiri, and M. Mohammadi, "Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty," *Transportation Research Part E: Logistics and Transportation Review*, vol. 89, pp. 182–214, 2016.
- [32] K. Sastry, D. Goldberg, and G. Kendall, "Genetic algorithms," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, pp. 97–125, Springer, 2005.
- [33] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, pp. 10–15, 1993.
- [34] L. Wang and D. Tang, "An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7243–7250, 2011.
- [35] S. S. Barreto, "Location-routing problems (LRP)," http://sweet.ua.pt/~iscf143/_private/SergioBarretoHomePage.htm.

Research Article

Regression and ANN Models for Electronic Circuit Design

M. I. Dieste-Velasco , M. Diez-Mediavilla , and C. Alonso-Tristán 

Electromechanical Engineering Department, University of Burgos, Burgos, Spain

Correspondence should be addressed to M. I. Dieste-Velasco; midieste@ubu.es

Received 29 December 2017; Accepted 17 May 2018; Published 16 July 2018

Academic Editor: Laszlo T. Koczy

Copyright © 2018 M. I. Dieste-Velasco 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 presents a methodology to design and to predict the behaviour of electronic circuits, which combines artificial neural networks and design of experiments. This methodology can be used to model output variables in electronic circuits either with similar features to the circuit configuration that is analysed in this study or with more complex configurations in order to improve the process of electronic circuit design.

1. Introduction

Artificial neural networks (ANNs) are employed in a wide range of applications for optimization, system identification, control and pattern recognition, among others [1]. ANNs have been widely used in several fields of both engineering and technology, and many researchers have analysed their application over the past few years. Among the published research works in the field of ANNs, the research study of Kamar et al. [2] predicted the cooling load, compressor power input, and the performance coefficients of an air-conditioning system. Their results showed correlation coefficients that were very close to unity, which indicates that their ANN models predicted the selected performance parameters to a high degree of accuracy [2]. In another study, Xiao et al. [3] used backpropagation neural networks for predicting oxygen dissolved in water with a combination of linear, log-sig, and tansig transfer functions. The authors found that the neural networks yielded the most accurate results, in comparison with common prediction methods such as curve fitting and autoregression, among various others [3]. Further examples might include works by Peng et al. [4] in which short-term wind power produced at a wind farm was analysed with ANNs and a hybrid strategy based on both physical and statistical methods [4]. It is also worth mentioning the study in which Karami [5] proposed a method of estimating transient stability analysis in a power system using multilayer perceptrons. The author concluded that the proposed

approach was highly suitable for online normalized transient stability margin estimation, because of its accuracy and computational efficiency [5]. Moreover, Srinivasan and Saghir [6] reported the results of modelling thermodiffusion in molten metals, by training neural networks with the Levenberg-Marquardt backpropagation algorithm. They concluded that a well-trained neural network can be reliably employed to quantify the thermotransport properties of binary metal alloys [6]. On the other hand, in the research study of Notton et al. [7], the authors determined global solar irradiation on tilted planes, by using multilayer perceptron, feedforward backpropagation, and the Levenberg-Marquardt algorithm [7]. In other studies, Gao et al. [8] proposed a modelling and error compensation approach based on backpropagation neural networks, for an articulated arm coordinate measuring machine. Likewise, Pedro et al. [9] published their design of a neural network for an antilock braking system. They concluded that the NN-based controller provided better results than the generic PID-based controller [9].

In addition to artificial neural networks, fuzzy set theory and their combination have been studied in depth over the past few years [10] and many researchers have investigated their application. For instance, Lovassy et al. [10] proposed fuzzy flip-flop-based neural networks (FNN) as an implementation of multilayer perceptron, in which the network was trained with the Levenberg-Marquardt algorithm. These authors found that this neural network is able to approximate both single and multiple variable functions to a high degree

of accuracy [10]. Zhang et al. [11] proposed the application of a hierarchical fuzzy neural network as a new type of fault diagnosis for an asynchronous motor, which performed effective diagnosis of both single and multiple faults [11]. Yang et al. [12] employed FNNs to provide an efficient means of detecting and evaluating concrete strength, and Yang et al. [13] combined fuzzy theory and artificial neural network techniques to evaluate and to analyse the safety status of an oil depot [13]. In further research studies, Lovassy et al. [14] reported an analysis of the behaviour of an FPGA implementation of the FNNs and went on to state that their FPGA implementation provided a safe and hazardless solution in environments where observation might be distorted by a component of noise [14].

In other research studies that concern the applicability of ANNs, Djefal et al. [15] showed the applicability of ANNs in the simulation of nanoscale CMOS circuits, training the ANN with a backpropagation algorithm. Moreover, Mohagheghi et al. [16] employed neural networks for the monitoring of electronic power circuits. Among their conclusions, they mentioned that, with a proper and a systematic selection of design parameters and sufficient training data, a neural network can learn to model any nonlinear system to a high level of accuracy [16].

This present research study compares the results provided by ANNs with those of a conventional regression. Moreover, it examines how the ANN is employed to obtain the main effect plots and the interaction effect plots, in order to determine the influence of the input parameters on the circuit response. It is shown that ANNs can be used to analyse electronic circuits and to optimize their performance. This methodology may also be used to model other output variables in electronic circuits with more complex configurations, in order to improve the process of electronic circuits design.

2. Modelling of Circuit Parameters

A bipolar transistor, configured as a common emitter amplifier, is selected as an example application to show the proposed methodology. The gain voltage and the relation between the collector current and base current (I_C/I_B) of the transistor are determined, following a full factorial design of experiments when the electrical resistances in the circuit are varied, from software simulations. These output values are then used to train an artificial neural network, in order to model both the gain voltage and the I_C/I_B as a function of the electrical resistances in the amplifier circuit. Once the artificial neural network is trained, it can then be used to obtain both the gain voltage and the I_C/I_B , when the electrical resistances are varied, with no further need for additional simulations. A 3^4 factorial design of experiments (DOE) was selected, which corresponds to 81 possible combinations of the input parameters (electrical resistances).

Table 1 shows the range of variation of the electrical resistances that form part of the electrical circuit shown in Figure 1. This circuit is analysed in this present study by training a neural network and then using the results to obtain the main effect plots and the interaction effect plots. That is,

TABLE 1: Electrical resistance values.

	Low	Nominal	High
R_1 (k Ω)	9.00	12.0	15.0
R_2 (k Ω)	2.25	3.0	3.8
R_3 (k Ω)	3.00	4.0	5.0
R_4 (k Ω)	1.50	2.0	2.5

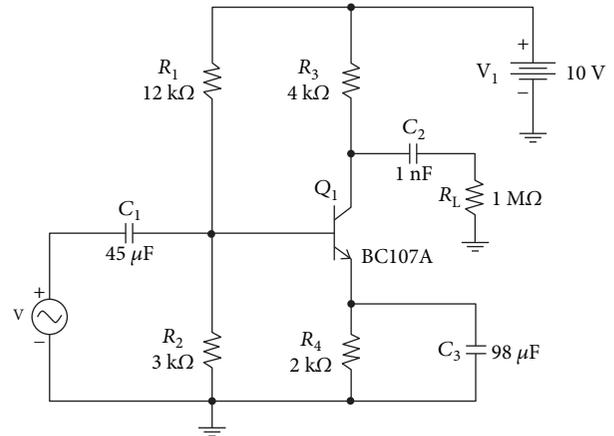


FIGURE 1: Nominal values of the amplifier circuit.

Table 1 shows the electrical resistance levels, which were selected following a full factorial DOE, based on raising and lowering the nominal value of the electrical resistances by 25% (from the centre value). The number of resistance values which could be employed, following this methodology, may be higher than three with the aim of obtaining a large number of data to train the ANN. Nevertheless, in this study, we have considered that three levels of variation is a high enough number to obtain significant results.

Figure 1 shows the nominal values of the amplifier circuit employed in this study, where R_j are the electrical resistances to be varied and V is a 20 mV and 1 kHz sinusoidal voltage source.

The values shown in Figure 1 may be modified with the data shown in Table 1, following a full-factorial DOE. After simulation of these electronic circuits, it will be possible to determine the gain voltage and the relation between both the collector and base currents of the transistor.

3. ANN and Regression Modelling

As was previously mentioned, simulations made it possible to determine the output data that correspond to all of the possible combinations of electrical resistances shown in Table 1. This will lead to different transistor polarizations that will induce variations in the gain voltage and in the I_C/I_B . By using the data shown in Table 2, a neural network can then be trained to predict both the I_C/I_B and the gain voltage of the amplifier circuit.

The ANN employed in this present study is composed of a hidden layer and an output layer. Figure 2 shows the configuration of the neural network used for modelling the

TABLE 2: Gain voltage $\Delta(V)$ and I_C/I_B results.

n	R_1	R_2	R_3	R_4	$\frac{I_C}{I_B}$	$\Delta(V)$
1	15.00	3.00	4.00	2.50	149.41	2.33
2	15.00	2.25	5.00	1.50	150.66	3.12
3	9.00	2.25	4.00	1.50	162.94	5.04
4	12.00	3.80	4.00	1.50	166.54	6.24
5	15.00	3.00	3.00	2.50	149.84	1.76
6	12.00	2.25	5.00	1.50	156.84	4.52
7	9.00	2.25	5.00	1.50	162.25	6.29
8	9.00	3.80	3.00	2.50	163.44	3.93
9	9.00	3.80	5.00	2.00	164.41	5.36
10	15.00	3.80	5.00	2.00	157.53	4.83
11	15.00	3.80	3.00	2.50	154.99	2.33
12	9.00	3.80	3.00	1.50	171.68	5.78
13	15.00	3.80	4.00	2.00	158.30	3.82
14	9.00	3.80	3.00	2.00	167.19	4.84
15	9.00	3.00	3.00	2.00	163.91	3.87
16	15.00	3.80	4.00	1.50	163.10	5.05
17	12.00	3.80	4.00	2.50	158.32	3.98
18	15.00	3.80	5.00	1.50	162.04	6.18
19	15.00	3.00	3.00	2.00	153.98	2.17
20	9.00	2.25	3.00	1.50	163.98	3.82
21	9.00	3.00	4.00	2.50	159.14	4.19
22	9.00	2.25	4.00	2.00	158.11	3.84
23	9.00	2.25	5.00	2.00	157.36	4.77
24	12.00	3.00	3.00	2.00	158.84	2.85
25	12.00	2.25	5.00	2.50	147.51	2.68
26	12.00	2.25	5.00	2.00	151.44	3.34
27	15.00	2.25	5.00	2.00	145.60	2.38
28	12.00	2.25	4.00	1.50	157.04	3.48
29	9.00	3.00	4.00	2.00	162.84	5.11
30	9.00	3.80	5.00	2.50	161.27	5.70
31	9.00	3.00	5.00	1.50	165.76	5.68
32	9.00	3.80	4.00	2.00	165.80	5.79
33	12.00	2.25	3.00	1.50	157.72	2.63
34	15.00	2.25	4.00	1.50	151.12	2.50
35	9.00	2.25	3.00	2.50	154.78	2.32
36	9.00	2.25	5.00	2.50	153.60	3.86
37	12.00	2.25	3.00	2.00	152.44	2.00
38	12.00	2.25	4.00	2.00	151.94	2.65
39	15.00	3.00	5.00	2.00	152.88	3.60
40	12.00	3.00	3.00	2.50	154.75	2.30
41	15.00	3.00	5.00	2.50	148.98	2.92
42	12.00	3.00	3.00	1.50	163.94	3.74
43	12.00	3.00	5.00	2.50	153.58	3.85
44	12.00	3.80	3.00	1.50	167.94	4.85
45	15.00	2.25	3.00	2.00	146.28	1.44
46	12.00	3.00	4.00	2.50	154.17	3.09
47	12.00	2.25	3.00	2.50	148.28	1.61
48	12.00	3.80	5.00	2.00	161.05	6.02
49	9.00	2.25	3.00	2.00	158.87	2.85

TABLE 2: Continued.

n	R_1	R_2	R_3	R_4	$\frac{I_C}{I_B}$	$\Delta(V)$
50	12.00	3.80	5.00	1.50	165.15	5.96
51	9.00	3.00	3.00	2.50	159.98	3.16
52	15.00	2.25	3.00	1.50	151.59	1.89
53	12.00	2.25	4.00	2.50	147.89	2.14
54	9.00	3.80	5.00	1.50	145.53	4.83
55	15.00	2.25	4.00	2.00	145.94	1.91
56	15.00	2.25	4.00	2.50	141.87	1.55
57	12.00	3.80	3.00	2.00	163.07	3.65
58	12.00	3.00	5.00	2.00	157.33	4.76
59	12.00	3.00	4.00	1.50	162.90	5.03
60	12.00	3.80	5.00	2.50	157.53	4.93
61	15.00	2.25	5.00	2.50	141.61	1.93
62	9.00	3.00	3.00	1.50	168.73	5.14
63	9.00	3.00	5.00	2.00	161.76	6.10
64	15.00	3.00	4.00	1.50	158.47	3.78
65	9.00	3.00	5.00	2.50	158.30	5.09
66	12.00	3.80	4.00	2.00	162.06	4.91
67	15.00	3.80	3.00	1.50	164.15	3.78
68	15.00	3.00	3.00	1.50	159.23	2.85
69	9.00	3.80	4.00	1.50	169.76	5.33
70	9.00	2.25	4.00	2.50	154.19	3.07
71	15.00	3.80	5.00	2.50	153.80	3.91
72	15.00	3.00	5.00	1.50	157.72	4.76
73	12.00	3.00	5.00	1.50	158.61	6.12
74	15.00	3.80	3.00	2.00	159.06	2.88
75	9.00	3.80	4.00	2.50	162.35	5.16
76	15.00	3.00	4.00	2.00	153.43	2.88
77	15.00	2.25	3.00	2.50	142.14	1.16
78	9.00	3.00	4.00	1.50	167.24	6.15
79	12.00	3.00	4.00	2.00	158.08	3.83
80	15.00	3.80	4.00	2.50	154.39	3.09
81	12.00	3.80	3.00	2.50	159.11	2.96

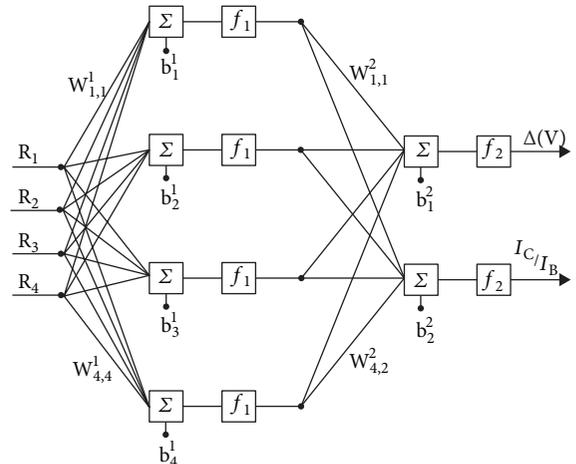


FIGURE 2: ANN structure.

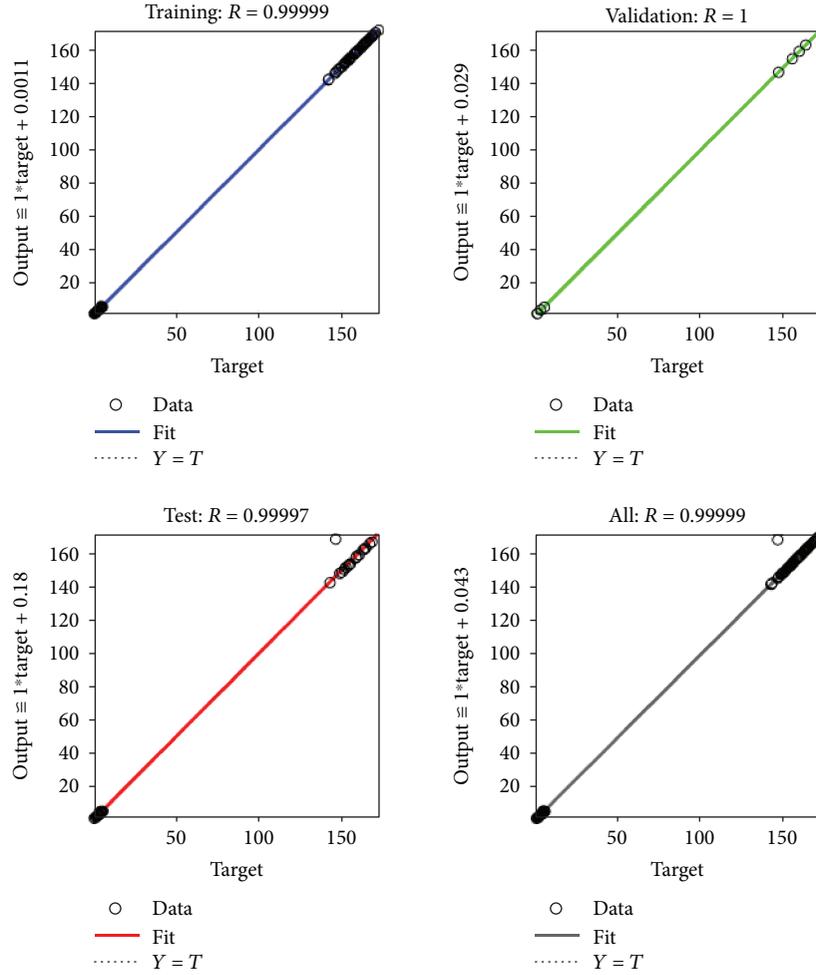


FIGURE 3: Results using the ANN.

behaviour of both the I_C/I_B and the $\Delta(V)$. The results dealing with the ANN were calculated with the Neural Network Toolbox™ of MATLAB® (MathWorks Inc.).

As may be seen, the same ANN was employed to model both outputs. Following a trial-and-error process, a hidden layer with four neurons was selected for modelling the values of both I_C/I_B and $\Delta(V)$.

Figure 2 shows the neural network configuration, which corresponds to four inputs ($R_1, R_2, R_3,$ and R_4), four neurons in the hidden layer, and two outputs (I_C/I_B and $\Delta(V)$). A DOE was used to train the ANN network, in order to obtain the main effect plot and the interaction effect plot. MSE was used as a performance index.

A log-sigmoid, defined by (1), was used as the transfer function (f_1) for the hidden layer, and a pure linear function, shown in (2), was used for the output layer (f_2), while a Levenberg-Marquardt algorithm was used to train the neural network.

$$f_1(x) = \frac{1}{1 + e^{-x}}, \quad (1)$$

$$f_2(x) = x. \quad (2)$$

Equations (3) and (4) may be obtained from Figure 2. These equations allow us to determine both the I_C/I_B and the $\Delta(V)$ as a function of the input parameters. As function f_2 is purely linear, (4) can be written as follows:

$$\text{output}_1 = f_1(W_1 * \text{inputs} + b_1), \quad (3)$$

$$\text{output}_2 = W_2 * f_1(W_1 * \text{inputs} + b_1) + b_2. \quad (4)$$

As previously mentioned, simulation results shown in Table 2 were used to train, to validate, and to test the neural network (70%, 15%, and 15%), respectively. Figure 3 shows the values obtained to measure the correlation between outputs and targets using the ANN. The performance index is also shown in Figure 4.

Figure 4 shows the training performance as well as the error histogram and the regression plot for the ANN considered in this present study. A conventional regression analysis was also performed using the whole set of experiments. The adjusted R -squared statistic indicated that the regression model explained 88% of the variability in I_C/I_B and 90% of the variability in $\Delta(V)$. As can be observed, these values are much smaller than those obtained with the ANN.

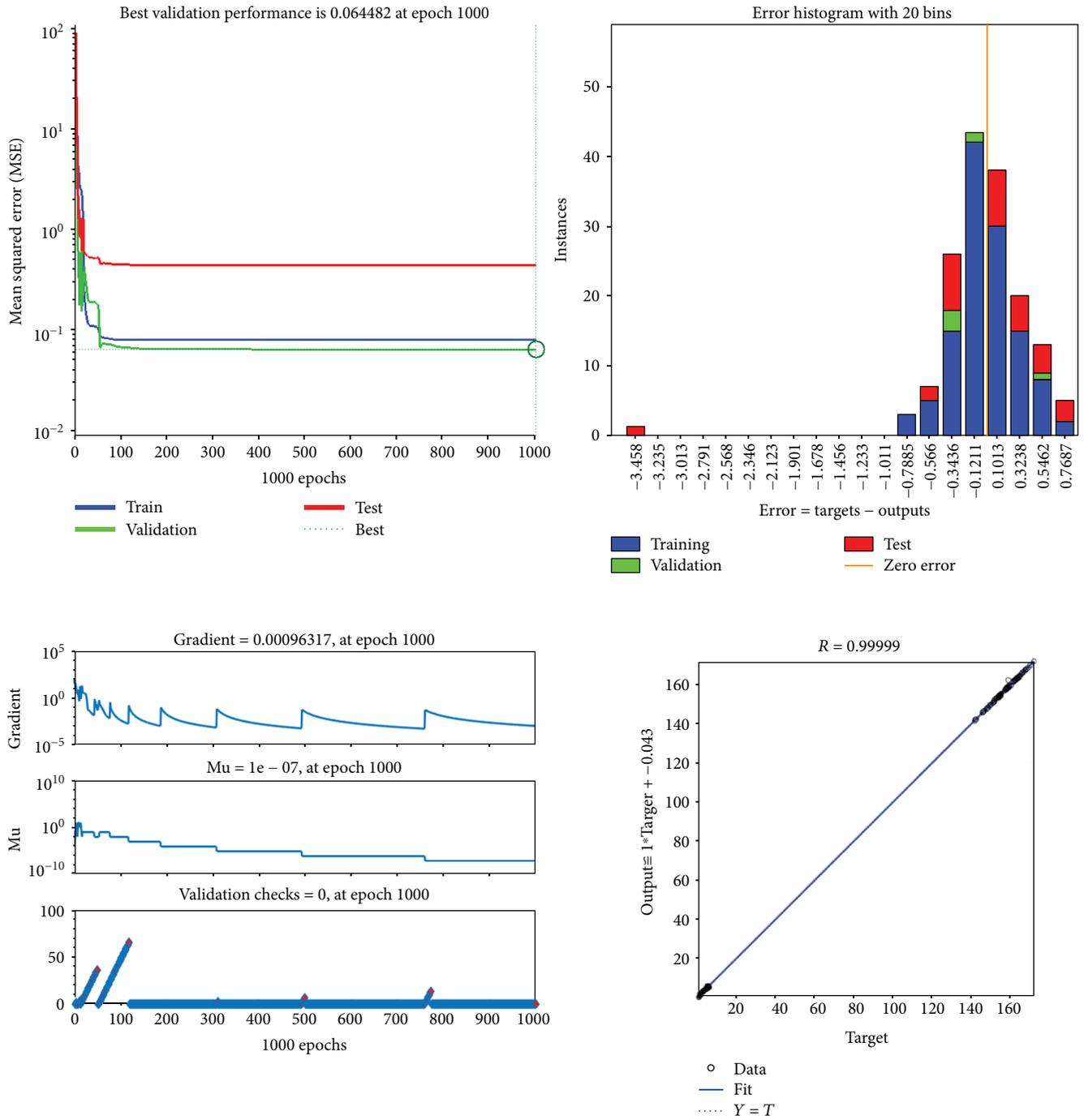


FIGURE 4: Neural network training state and error diagrams.

TABLE 3: First layer weights and biases.

Weights				Biases
-0.3842	0.2583	-0.4092	-0.4423	-0.2399
-0.3798	0.2564	-0.3915	-0.4369	-0.2456
-0.8592	1.6675	0.0988	-0.3028	2.3048
3.7471	-3.5843	-3.7366	3.1795	18.8845

TABLE 4: Second layer weights and biases.

Weights				Biases
-57.2214	60.3744	0.9869	164.7291	-166.7892
-174.2275	178.5871	0.9639	81.1056	-83.5416

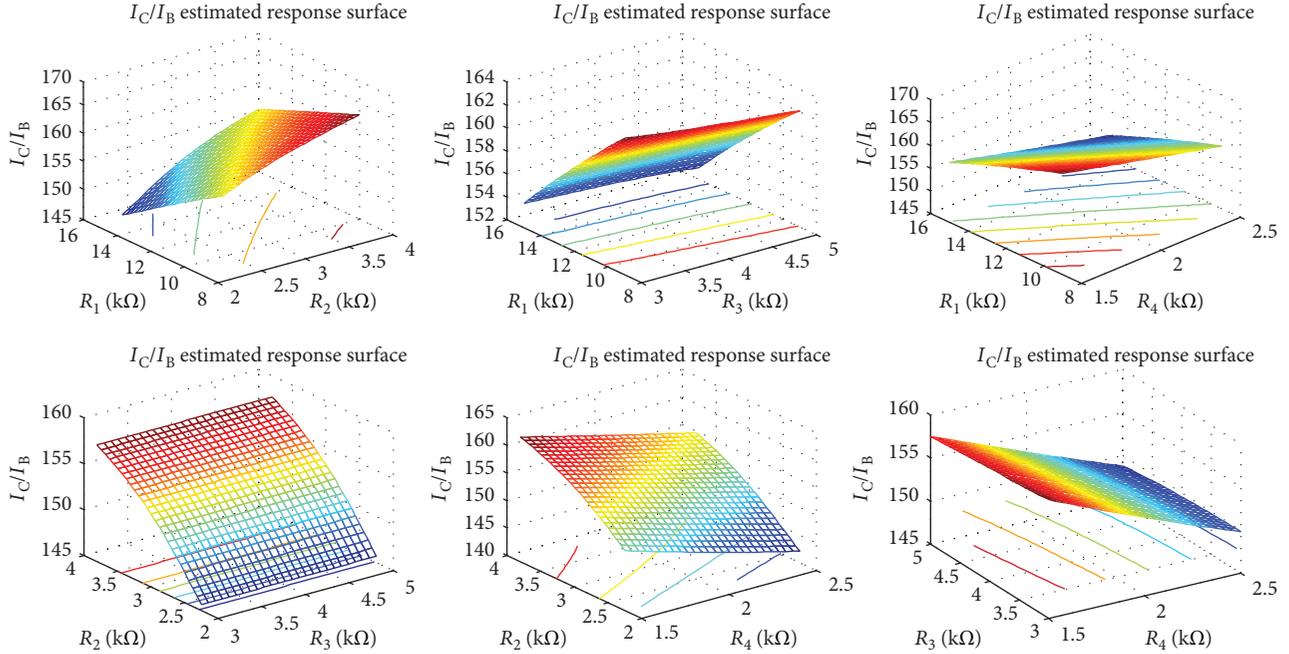


FIGURE 5: Estimated ANN response surface for the I_C/I_B .

Equation (5) show the regression polynomials for the I_C/I_B and the $\Delta(V)$.

$$\begin{aligned}
 \frac{I_C}{I_B} = & 187.197 - 3.145 * R_1 + 12.026 * R_2 - 1.281 \\
 & * R_3 - 14.881 * R_4 + 0.781 * R_1 * R_2 + 0.295 \\
 & * R_1 * R_3 - 0.560 * R_1 * R_4 - 1.192 * R_2 * R_3 \\
 & + 2.275 * R_2 * R_4 + 1.848 * R_3 * R_4 - 0.031 \\
 & * R_1^2 - 2.530 * R_2^2 - 0.457 * R_3^2 - 0.085 * R_4^2, \quad (5)
 \end{aligned}$$

$$\begin{aligned}
 \Delta(V) = & 3.093 - 0.528 * R_1 + 2.055 * R_2 + 1.480 * R_3 \\
 & - 2.840 * R_4 + 0.086 * R_1 * R_2 + 0.013 * R_1 \\
 & * R_3 - 0.025 * R_1 * R_4 - 0.018 * R_2 * R_3 \\
 & + 0.072 * R_2 * R_4 + 0.020 * R_3 * R_4 - 0.001 \\
 & * R_1^2 - 0.335 * R_2^2 - 0.108 * R_3^2 + 0.344 * R_4^2.
 \end{aligned}$$

Tables 3 and 4 show the weights and biases of the ANN. Once determined, the ANN may be used to predict the output values. Moreover, it is possible to employ the ANN to obtain response surfaces for the I_C/I_B and the $\Delta(V)$ as well as to analyse the effect of the input variables on these parameters. As previously shown, the results of the neural network are of higher accuracy than the results of the regression analysis. The ANN therefore has many advantages over traditional methodologies that analyse the DOE from mean values. However, the models obtained with the ANN are often more complex than those obtained by regression analysis.

4. Discussion

Figures 5 and 6 show the neural network response surfaces for both the I_C/I_B and the $\Delta(V)$. Once the weight and bias parameters are obtained, it is possible to plot

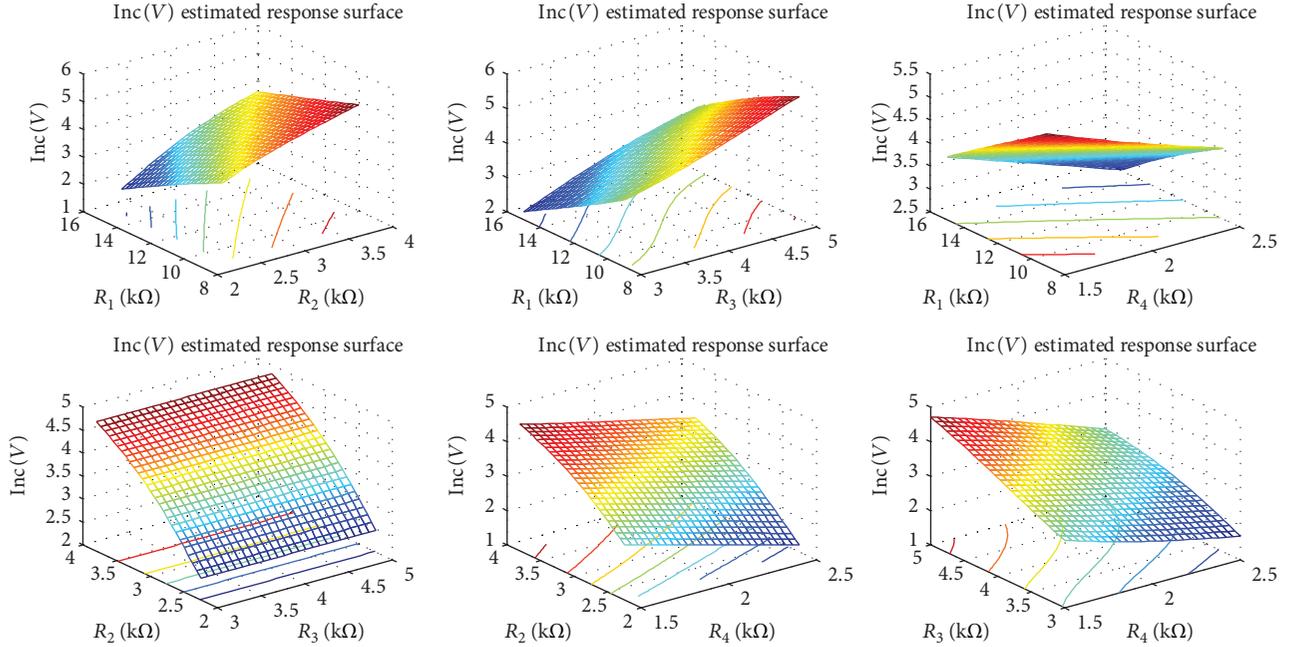
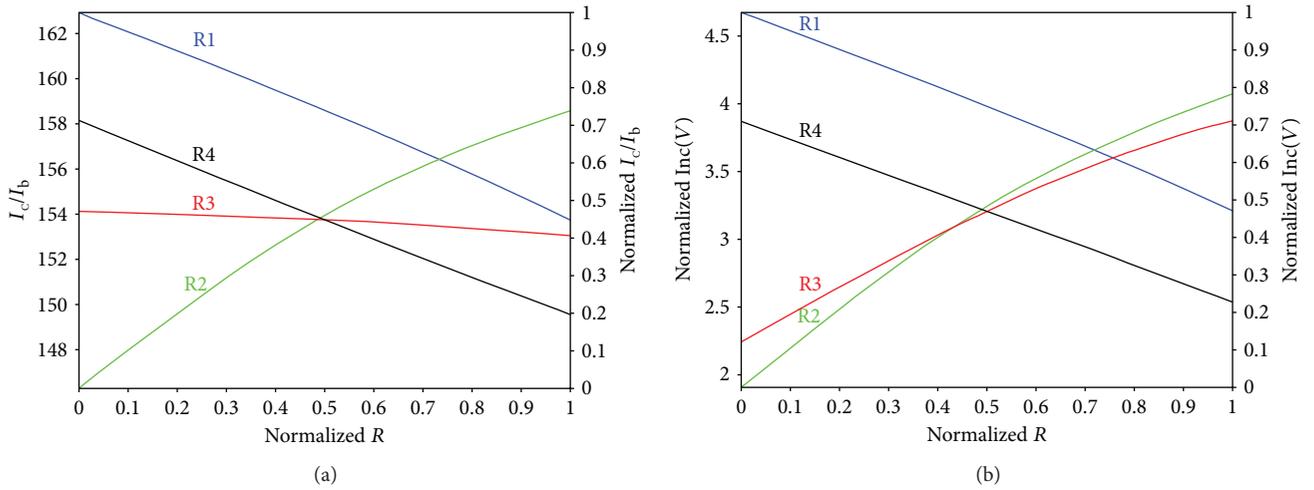
the response surfaces. For example, the response surface $I_C/I_B = f(R_1, R_2)$ is obtained when both R_1 and R_2 are varied, while R_3 and R_4 are kept at their central values.

The same procedure can be used to determine the main effect plot and the interaction effect plot. In this case, to obtain the main effect plot, one input parameter is varied while the others are kept at their central values, as shown by Figure 7. The interaction plot between two parameters is obtained by considering the higher and the lower values of one parameter, while the other is modified in its variation range and the rest of the parameters are kept at their central values, as shown in Figures 8 and 9. In this way, it is possible to obtain the interaction effect plot and the main effect plot using the ANN, where the inputs have to be selected according to the aforementioned procedure.

As mentioned earlier, the influence of the parameters on both the I_C/I_B and the $\Delta(V)$ may be analysed by means of the interaction effect plot and the main effect plot. These plots can be obtained by determining the outputs provided by the neural network when appropriate inputs are considered.

Figure 7 shows the main effect plot which has been normalized from the expressions shown in (6), in which the response variables were obtained from the data given by the previously determined artificial neuronal network.

$$\begin{aligned}
 \frac{I_C}{I_{b_{\text{normalized}_{R_j}}}} = & \frac{I_C/I_{b_{R_j}} - \min(I_C/I_{b_{R_1, \dots, R_4}})}{\max(I_C/I_{b_{R_1, \dots, R_4}}) - \min(I_C/I_{b_{R_1, \dots, R_4}})}, \\
 \Delta(V)_{\text{normalized}_{R_j}} = & \frac{\Delta(V)_{R_j} - \min(\Delta(V)_{R_1, \dots, R_4})}{\max(\Delta(V)_{R_1, \dots, R_4}) - \min(\Delta(V)_{R_1, \dots, R_4})},
 \end{aligned}$$

FIGURE 6: Estimated ANN response surface for the $\Delta(V)$.FIGURE 7: Main effect plots showing the I_C/I_B and the $\Delta(V)$ values obtained with the ANN.

$$\text{normalized}_{R_j} = \frac{R_j - \min(R_j)}{\max(R_j) - \min(R_j)}. \quad (6)$$

As can be observed in Figure 7(a), for this present study, when R_2 – R_4 are kept at their central values, the I_C/I_B reaches a maximum value, if R_1 is kept at its minimum value. The same behaviour is observed for R_3 and R_4 . On the contrary, I_C/I_B reaches a maximum value, if R_2 is kept at its maximum value, while R_1 , R_3 , and R_4 are kept at their central values.

As may also be observed in Figure 7(a), R_2 is the parameter which affects the I_C/I_B more than any other, followed by R_1 and R_4 . The collector resistance (R_3) parameter affects the

I_C/I_B less than any other, and its influence is much smaller than the influence of the other three electrical resistances.

In the case of the gain voltage, as can also be observed in Figure 7(b), the value of $\Delta(V)$ reaches a maximum, when R_2 – R_4 are kept at their central values, if R_1 is kept at its minimum value. The same behaviour may be observed for R_4 . On the contrary, the value of $\Delta(V)$ reaches a maximum, if either R_2 or R_3 is kept at its maximum value, while the rest of the electrical resistances are kept at their central values. As may also be observed in Figure 7(b), the most influential parameters are R_2 and R_3 , followed by R_1 and R_4 .

As can be observed in Figure 8, R_2 exhibits a different behaviour, with respect to R_1 and to the behaviour shown by R_3 and R_4 . The same trend is observed in the three electrical resistors R_2 – R_4 , when they are kept at either minimum or

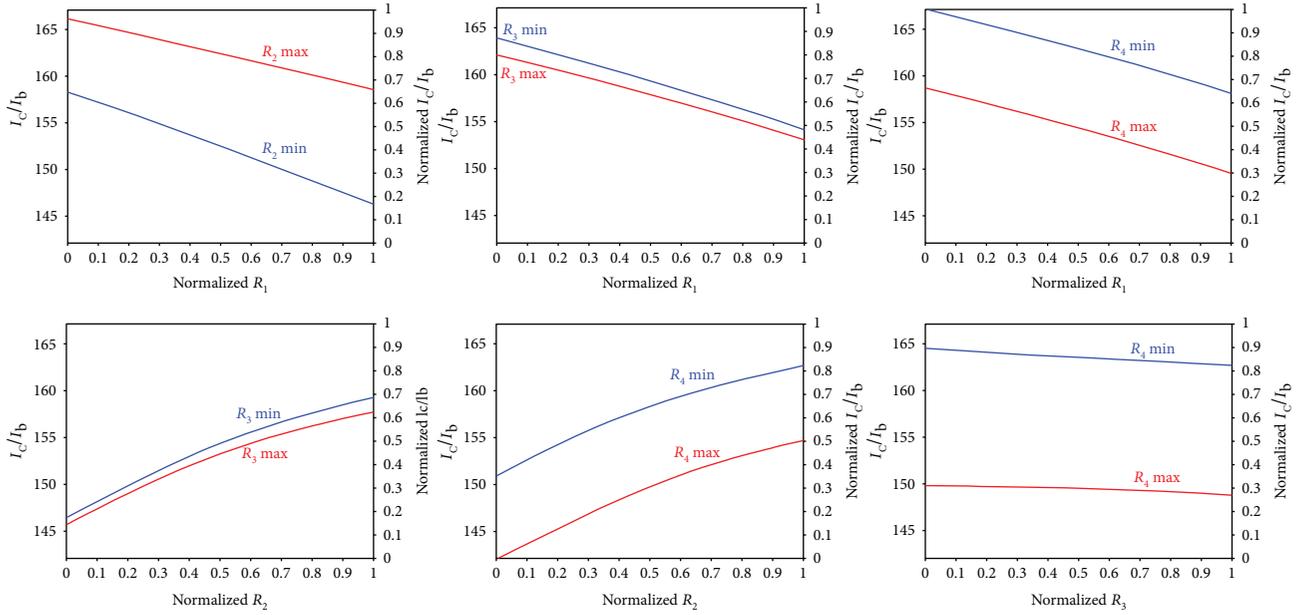


FIGURE 8: Interaction effect plot showing the I_C/I_B values obtained with the ANN.

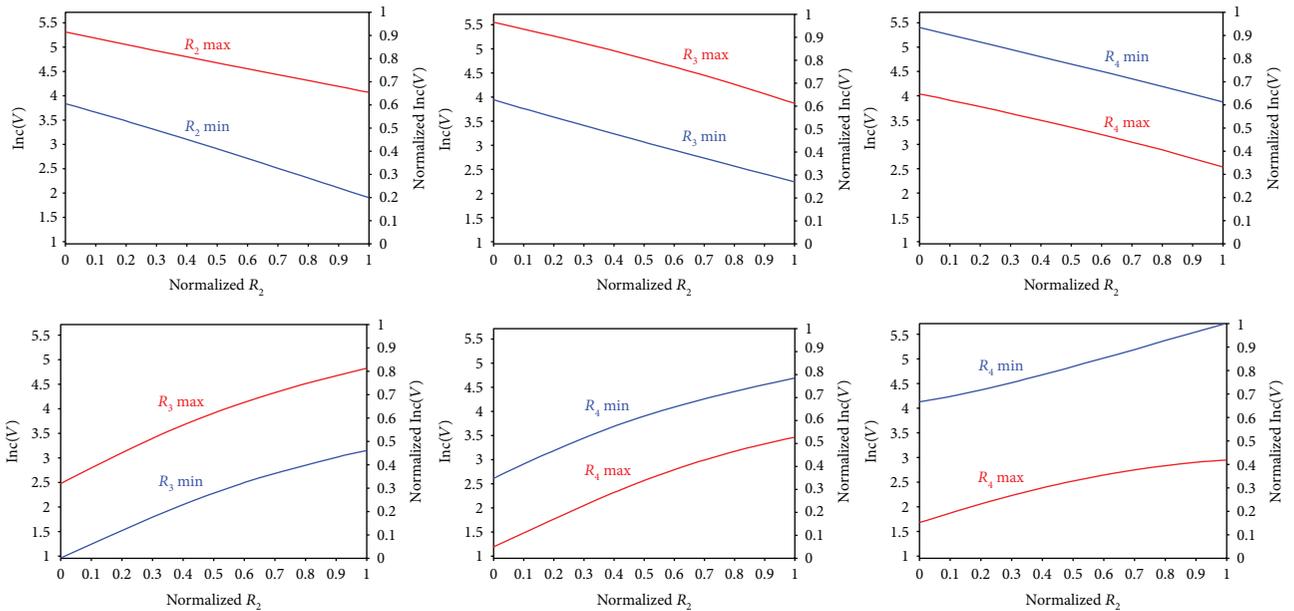


FIGURE 9: Interaction effect plots for the $\Delta(V)$ obtained with the ANN.

maximum values and R_1 is varied, that is, the lower the electrical resistances, the higher the I_C/I_B . The influence of R_3 and R_4 on R_2 is approximately the same, while the interaction between R_3 and R_4 is much lower.

As can be observed in Figure 9, the behaviour of R_4 in relation to R_1 differs in relation to R_2 and R_3 . The same trend is observed in the three electrical resistors R_2 – R_4 , when they are kept at either minimum or maximum values, while R_1 is varied: the lower the electrical resistance, the higher the $\Delta(V)$. The influence of R_3 and R_4 on R_2 is very similar, but its behaviour is different, the interaction between the collector resistance (R_3) and the emitter resistance (R_4) being much

lower than the other interactions between the electrical resistances of the amplifier circuit.

5. Conclusions

In this present study, a neural network comprising four inputs, a hidden layer of four neurons, and two outputs has been used to model output variables in an electronic circuit. The findings have shown that a factorial DOE combined with an ANN can model the behaviour of I_C/I_B and $\Delta(V)$ in an efficient way. Higher accuracy was obtained with the ANN than with conventional regression techniques.

The methodology proposed in this study therefore has the potential to assist the analytical design of electronic circuits and their actual implementation, either with similar features to that shown in this present study or with more complex configurations.

Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

References

- [1] Y. Jiang, C. Yang, J. Na, G. Li, Y. Li, and J. Zhong, "A brief review of neural networks based learning and control and their applications for robots," *Complexity*, vol. 2017, Article ID 1895897, 14 pages, 2017.
- [2] H. M. Kamar, R. Ahmad, N. B. Kamsah, and A. F. Mohamad Mustafa, "Artificial neural networks for automotive air-conditioning systems performance prediction," *Applied Thermal Engineering*, vol. 50, no. 1, pp. 63–70, 2013.
- [3] Z. Xiao, L. Peng, Y. Chen, H. Liu, J. Wang, and Y. Nie, "The dissolved oxygen prediction method based on neural network," *Complexity*, vol. 2017, Article ID 4967870, 6 pages, 2017.
- [4] H. Peng, F. Liu, and X. Yang, "A hybrid strategy of short term wind power prediction," *Renewable Energy*, vol. 50, pp. 590–595, 2013.
- [5] A. Karami, "Power system transient stability margin estimation using neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 4, pp. 983–991, 2011.
- [6] S. Srinivasan and M. Z. Saghir, "Modeling of thermotransport phenomenon in metal alloys using artificial neural networks," *Applied Mathematical Modelling*, vol. 37, no. 5, pp. 2850–2869, 2013.
- [7] G. Notton, C. Paoli, L. Ivanova, S. Vasileva, and M. L. Nivet, "Neural network approach to estimate 10-min solar global irradiation values on tilted planes," *Renewable Energy*, vol. 50, pp. 576–584, 2013.
- [8] G. Gao, H. Zhang, H. San, X. Wu, and W. Wang, "Modeling and error compensation of robotic articulated arm coordinate measuring machines using BP neural network," *Complexity*, vol. 2017, Article ID 5156264, 8 pages, 2017.
- [9] J. O. Pedro, O. T. C. Nyandoro, and S. John, "Neural network based feedback linearisation slip control of an anti-lock braking system," in *Proceedings of the 7th Asian Control Conference*, pp. 1251–1257, Hong Kong, China, August 2009.
- [10] R. Lovassy, L. T. Kóczy, and L. Gál, "Function approximation capability of a novel fuzzy flip-flop based neural network," in *Proceedings of the 2009 International Joint Conference on Neural Networks*, pp. 1900–1907, Atlanta, GA, USA, June 2009.
- [11] Y. Zhang, N. Xu, and Q. Ji, "Study on the application of the hierarchical fuzzy neural network in the fault diagnosis of the asynchronous motor," in *2010 Sixth International Conference on Natural Computation (ICNC)*, pp. 651–654, Yantai, China, August 2010.
- [12] S. S. Yang, J. Xu, and G. Z. Yao, "Concrete strength evaluation based on fuzzy neural networks," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, pp. 3344–3347, Shanghai, China, August 2004.
- [13] Y. Yang, C. Zhu, and J. Xing, "Application of fuzzy neural network to safety evaluation for oil depot," in *2012 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, pp. 55–58, Chengdu, China, October 2012.
- [14] R. Lovassy, L. Gál, Á. Tóth, L. T. Kóczy, and I. J. Rudas, "Fuzzy Flip-flop based neural networks as a novel implementation possibility of multilayer perceptrons," in *2012 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 280–285, Graz, Austria, May 2012.
- [15] F. Djeflal, M. Chahdi, A. Benhaya, and M. L. Hafiane, "An approach based on neural computation to simulate the nanoscale CMOS circuits: application to the simulation of CMOS inverter," *Solid-State Electronics*, vol. 51, no. 1, pp. 48–56, 2007.
- [16] S. Mohagheghi, R. G. Harley, T. G. Habetler, and D. Divan, "Condition monitoring of power electronic circuits using artificial neural networks," *IEEE Transactions on Power Electronics*, vol. 24, no. 10, pp. 2363–2367, 2009.

Research Article

Integrated Feature Selection of ARIMA with Computational Intelligence Approaches for Food Crop Price Prediction

Yuehjen E. Shao  and Jun-Ting Dai

Department of Statistics and Information Science, Fu Jen Catholic University, New Taipei City, Taiwan

Correspondence should be addressed to Yuehjen E. Shao; stat1003@mail.fju.edu.tw

Received 26 September 2017; Revised 22 May 2018; Accepted 3 June 2018; Published 10 July 2018

Academic Editor: Laszlo T. Koczy

Copyright © 2018 Yuehjen E. Shao and Jun-Ting Dai. 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.

Because of global climate change, lack of arable land, and rapid population growth, the supplies of three major food crops (i.e., rice, wheat, and corn) have been gradually decreasing worldwide. The rapid increase in demand for food has contributed to a continuous rise in food prices, which directly threatens the lives of over 800 million people around the world who are reported to be chronically undernourished. Consequently, food crop price prediction has attracted considerable attention in recent years. Recent integrated forecasting models have developed various feature selection methods (FSMs) to capture fewer, but more important, explanatory variables. However, one major problem is that the future values of these important explanatory variables are not available. Thus, predictions based on these variables are not actually possible. Because an autoregressive integrated moving average (ARIMA) can extract important self-predictor variables with future values that can be calculated, this study incorporates an ARIMA as the FSM for computational intelligence (CI) models to predict three major food crop (i.e., rice, wheat, and corn) prices. Other than the ARIMA, the components of the proposed integrated forecasting models include artificial neural networks (ANNs), support vector regression (SVR), and multivariate adaptive regression splines (MARS). The predictive accuracies of ARIMA, ANN, SVR, MARS, and the proposed integrated model are compared and discussed. Experimental results reveal that the proposed integrated model achieves superior forecasting performance for predicting food crop prices.

1. Introduction

1.1. Background. Everyone needs food. However, not everyone has enough food to survive. Food crops are a primary source of human food, with rice, wheat, and corn being the most widely consumed sources of grains around the world. In this study, food crops refer to plants, which provide food for human consumption, cultivated by agriculturists. The demand and consumption for food crops will rapidly increase in the future, propelled by a 2.3 billion person increase in global population and greater per capita incomes anticipated through the midcentury [1, 2]. According to the Food and Agriculture Organization of the United Nations (FAO) [3], the demand for food is expected to grow substantially by 2050. A major factor for this increase is world population growth. Today, world population has reached

7.6 billion, and we may reach 9.7 billion by 2050. This growth, along with rising incomes in developing countries, is driving up global food demand. Consequently, humanity may directly face one difficulty, which is “will enough food crops be produced at affordable prices or will rising prices drive more of the humanity into hunger?”

The three most important food crops, rice, wheat, and corn, directly provide more than 50% of all calories consumed by the world population. While the area harvested for wheat each year is 214 million ha, the areas harvested for rice and corn are 154 million ha and 140 million ha, respectively [4]. Additionally, the rapid rise in food prices has been a burden on the poor in developing countries, who spend roughly half their household income on food. The issue of being able to affordably purchase the aforementioned food crops plays a very important role in human life.

Therefore, from the human point of view, the prediction of prices for rice, wheat, and corn has become a significant research topic.

1.2. Related Work. Owing to various agricultural and environmental factors, meteorological factors, and biophysical factors, it was indicated in [5] that the exports of rice were very difficult to predict. They employed the autoregressive integrated moving average (ARIMA) and artificial neural network (ANN) methodologies to predict Thailand's rice exports. The wheat price in the Chinese market was predicted by using the ARIMA, ANN, and linear combination models [6]. The overall results showed that the ANN technique was the best prediction model. In Africa, around 40% of the rice consumed is imported [7]. This high dependence on rice imports indicates that Africa is highly exposed to international rice market shocks with sometimes grave consequences for its food security and political stability. In addition, models were constructed to predict the quarterly prices of two types of food crops, barley and wheat [8]. In [9], it was reported that the world food prices would rise around 32% by 2022.

They showed the dynamic relationship between acreage response, food crop yield, and price volatility by developing an optimization model [10]. In [11], it was stated that most countries would like to predict their annual food requirements in order to provide food security for the people. They proposed an artificial intelligent support vector regression (SVR) model to predict the output energy in rice production.

Because food crop prices are seasonally affected [5, 6], this study uses the ARIMA to predict the prices of rice, wheat, and corn. However, the assumptions inherent to the linear form of ARIMA may encounter problems in adopting nonlinear relationships for practical data [12]. Computational intelligence (CI) techniques have been widely used in many forecasting applications because of data-driven features and fewer a priori assumptions. Accordingly, in addition to ARIMA modeling, this study uses CI schemes, including ANN, SVR, and multivariate adaptive regression splines (MARS), for predicting the prices of the three food crops because they allow nonlinearity modeling and provide good forecasting characteristics.

However, CI modeling may face difficulties in its training process for designing an optimal topology owing to the use of a high number of input variables. Therefore, feature selection methods (FSMs) have been incorporated in order to reduce the number of explanatory or predictor variables [13, 14]. In this study, feature selection refers to the process of identifying a subset of relevant explanatory or predictor variables for use during forecasting model construction. This subset of variables contains fewer but more important input variables that aid in predicting the outcome.

Feature selection techniques have become a research hot spot in many forecasting applications. For example, in order to accurately predict wind speed, an SVR forecasting model with a feature selection procedure called phase space reconstruction has been proposed [15]. Additionally, a set of general ARIMA models was used for performance comparison. A hybrid forecasting model with a feature selection technique

based on mutual information, extreme learning machines, and bootstrap techniques was proposed to predict day-ahead electricity prices [16]. The authors of [17] proposed a hybrid filter-wrapper approach to predict electrical load and the price of electricity. The feature selection method used in their approach identifies a minimum subset of the most informative features by considering the relevancy, redundancy, and interactions of candidate inputs. In [18], the authors proposed a two-step approach that selects a set of candidate features based on data characteristics. A hybrid ANN-based model was then used to predict the day-ahead price of electricity. A hybrid forecasting methodology was designed to predict electrical load in [19]. The study used a feature selection method that was developed based on entropy and several CI techniques. The prediction performance of the proposed model was compared to various ARIMA models. A regression model was proposed to predict online sales in [20]. Additionally, a feature selection methodology based on a multiobjective evolutionary algorithm was combined with the forecasting model. A combined ANN and Kalman filter approach was proposed to predict wind speed in [21]. An ARIMA feature selection method was used to determine the input structure for their hybrid model.

When reviewing related research, we noticed that, although many studies have focused on using FSMs to obtain accurate forecasts, little attention has been devoted to the use of FSMs for food crop price prediction. Additionally, we encounter practical problems when using FSMs to predict food crop prices. As mentioned earlier, one of the major purposes for using an FSM is to select a subset of explanatory variables for further use in forecasting. However, the future values of the explanatory variables that are selected by the FSM are unknown. Relatively little research has attempted to address this problem. As a consequence, predictions cannot be computed even though the most important explanatory variables have been identified by effective FSMs. We propose an integrated forecasting model to remedy the problems caused by unknown explanatory variables. The proposed integrated model employs an ARIMA as an FSM to capture important self-predictor variables, which then serve as input variables for the ANN, SVR, and MARS models. Because the future values of a self-predictor variable can be computed based on its previous and current values, food crop price predictions can be obtained.

In this study, we propose four single-stage models and three integrated models for predicting food crop prices. A real monthly dataset was obtained, which contains the prices of rice, wheat, and corn from January 1990 to September 2015. This real dataset makes it possible to compare predictions of food crop prices using single-stage models and integrated models. The remainder of the study is organized as follows: Section 2 presents the modeling methodologies of different forecasting schemes. In Section 3, we describe the design structure for our single- and two-stage integrated models. Real data on the prices of three food crops are collected to verify the single-stage models and integrated models. Section 4 contains forecasting comparisons for four single-stage models and three integrated models. Some practical implications and other concerns are addressed in

Section 5. The final section summarizes our research findings and contains our conclusions.

2. Forecasting Methodologies

2.1. Autoregressive Integrated Moving Average. Because seasonal effects are possibly involved in food crop price forecasting, seasonal ARIMA modeling should be developed [22]. A seasonal ARIMA model can be described as follows:

$$\phi_p(B)\Phi_P(B^L)(1-B)^d(1-B^L)^D Z_t = \delta + \theta_q(B)\Theta_Q(B^L)a_t, \quad (1)$$

where

B is the backward shift operator, defined as $B^j y_t = y_{t-j}$;

L is the number of seasons in a year, and $L = 12$ for monthly data;

d is the values of nonseasonal difference transformations;

D is the values of seasonal difference transformations;

Z_t is the working series, which are stationary after fitting a suitable difference transformation from original time series Y_t ;

δ is an unknown constant;

a_t is the white noise at time t , which is independent and identical (iid) with normal distribution;

p is the order of nonseasonal autoregressive (AR) models;

P is the order of seasonal AR models;

q is the order of nonseasonal moving average (MA) models;

Q is the order of seasonal MA models;

$\phi_p(B)$ is a polynomial function for a nonseasonal AR model, defined as $\phi_p(B) = (1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)$;

$\Phi_P(B^L)$ is a polynomial function for a seasonal AR model, defined as $\Phi_P(B^L) = (1 - \Phi_{1,L} B^L - \Phi_{2,L} B^{2L} - \dots - \Phi_{P,L} B^{PL})$;

$\theta_q(B)$ is a polynomial function for a nonseasonal MA model, defined as $\theta_q(B) = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)$;

$\Theta_Q(B^L)$ is a polynomial function for a seasonal MA model, defined as $\Theta_Q(B^L) = (1 - \Theta_{1,L} B^L - \Theta_{2,L} B^{2L} - \dots - \Theta_{P,L} B^{QL})$.

The original nonstationary time series should be transformed into a stationary working series through differencing. Typically, the appropriate transformations can be performed by four combinations of d and D ; that is, $(d, D) = (0, 0)$, $(d, D) = (1, 0)$, $(d, D) = (0, 1)$, and $(d, D) = (1, 1)$, respectively. Once the stationary working series has been attained, we can observe the behavior of the sample autocorrelation function (ACF) and sample partial autocorrelation function (PACF) to determine the values of p , P , q , and Q for the seasonal ARIMA models.

They prefer using the maximum likelihood (ML) technique to obtain estimates for model parameters [22]. For the ML technique, the likelihood function is maximized via nonlinear least squares using Marquardt's method. Because the ML approach is more computationally expensive than conditional least squares (CLS) estimates, most computer packages employ CLS as a default approach. LS refers to the parameter estimate associated with the smallest sum of

squared errors (SSE). For the CLS approach, we assume that the number of past unobserved errors is zero. The data series y_t can be represented as follows:

$$y_t = \sum_{i=1}^{\infty} \pi_i y_{t-i} + a_t. \quad (2)$$

The π weights are calculated as follows:

$$\frac{\phi(B)}{\theta(B)} = 1 - \sum_{i=1}^{\infty} \pi_i B^i. \quad (3)$$

The CLS approach should produce parameter estimates that can minimize the following:

$$\text{SSE} = \sum_{t=1}^n (a_t^\wedge)^2 = \sum_{t=1}^n \left(y_t - \sum_{i=1}^{\infty} \pi_i^\wedge y_{t-i} \right)^2, \quad (4)$$

where the unobserved past values of y_t are set to 0 and π_i is computed from the estimates of the ARIMA model parameters ϕ and θ at each iteration.

After estimating the model parameters, the diagnostic checks for testing the lack of fit of the seasonal ARIMA model should be conducted. A logical way to check the adequacy of a seasonal ARIMA model is to analyze the residuals obtained from the underlying model. The Ljung-Box test was developed to examine whether the first K sample autocorrelations of residuals indicate adequacy of the model [23]. The null hypothesis for this test is that the first K autocorrelations are jointly zero; that is,

$$H_0 : \rho_1 = \rho_2 = \dots = \rho_k = 0. \quad (5)$$

The Ljung-Box statistics are described as

$$Q = n_d(n_d + 2) \sum_{l=1}^k (n_l - l)^{-1} r_l^2(\hat{a}), \quad n_d = n - d, \quad (6)$$

where

Q is the Ljung-Box statistics, and the asymptotic distribution of Q follows a chi-square distribution with $k - n_p$ degrees of freedom;

n is the number of observations;

d is the degree of nonseasonal differencing;

n_p is the number of parameters other than δ that must be estimated in the ARIMA model under consideration;

$r_l^2(\hat{a})$ is the square of the sample autocorrelation of the residuals at lag l .

The Ljung-Box test rejects the null hypothesis, which indicates that the underlying model has significant lack of fit if

$$Q > \chi_\alpha^2(k - n_p), \quad (7)$$

where $\chi_\alpha^2(k - n_p)$ is the chi-square distribution table value with $k - n_p$ degrees of freedom and significance level α .

2.2. Artificial Neural Networks. ANN is composed of a large number of highly interconnected processing elements, which are referred to as nodes or neurons, working in parallel to

solve a particular problem. While the leftmost layer of the network is referred to as the input layer, the rightmost layer is called the output layer. The middle layer is called the hidden layer. In the ANN structure, each layer consists of a number of nodes connected by links. The ANN contains nodes that are connected to themselves, enabling a node to influence other nodes and itself. ANN includes input variables and output variables. In addition, a certain number of nodes are contained in the hidden layer, and the hidden nodes are nonlinear functions of the input variables. The output variable is a function of the nodes in the hidden layer.

ANN modeling is briefly described as follows. For an ANN model, the relationship between inputs ($\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_a$) and output (O) can be represented as

$$O = \alpha_0 + \sum_{j=1}^b \alpha_j g \left(\delta_{0j} + \sum_{i=1}^a \delta_{ij} \mathbf{I}_i \right) + \varepsilon, \quad (8)$$

where α_j ($j = 0, 1, 2, \dots, b$) and δ_{ij} ($i = 0, 1, 2, \dots, a; j = 0, 1, 2, \dots, b$) are the model connection weights; a is the number of input nodes; b is the number of hidden nodes; and ε is the error term. In the hidden layer, the transfer function is often used with a logistic function.

$$g(z) = \frac{1}{1 + \exp(-z)}. \quad (9)$$

Consequently, the ANN transports nonlinear functional mapping from the inputs ($\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_a$) to the output O ; namely,

$$O = f(\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_a, \mathbf{w}) + \varepsilon, \quad (10)$$

where \mathbf{w} is a vector of model parameters and f is a function determined by the ANN structure and connection weights.

For the ANN structure, the nodes in the input layers receive input signals from an external source and the nodes in the output layers generate the target output signals. The output of each neuron in the input layer is the same as the input to that neuron. The hidden layers adjust the weights of those inputs until the neural network's error is minimized. For ANN processing, backpropagation is one method for computing the error contribution of each neuron after a batch of data is processed. This method can be used to adjust the weight of each neuron, thereby completing the learning process for that case. For each neuron j in the hidden layer and each neuron k in the output layer, the net inputs are given by

$$\begin{aligned} \text{net}_j &= \sum_i w_{ji} \times o_i, \\ \text{net}_k &= \sum_j w_{kj} \times o_j, \end{aligned} \quad (11)$$

where $i(j)$ is a neuron in the previous layer, $o_i(o_j)$ is the output of node $i(j)$, and $w_{ji}(w_{kj})$ is the connection weight from neuron $i(j)$ to neuron $j(k)$. The neuron outputs are given by

$$\begin{aligned} o_i &= \text{net}_i, \\ o_j &= \frac{1}{1 + e^{-(\text{net}_j + \theta_j)}} = f_j(\text{net}_j, \theta_j), \\ o_k &= \frac{1}{1 + e^{-(\text{net}_k + \theta_k)}} = f_k(\text{net}_k, \theta_k), \end{aligned} \quad (12)$$

where $\text{net}_j(\text{net}_k)$ is the input signal from the external source to node $j(k)$ in the input layer and $\theta_j(\theta_k)$ is a bias.

The generalized delta rule is the conventional technique used to derive the connection weights in the network. First, a set of random numbers is assigned to the connection weights. Then for a presentation of a pattern p with a target output vector $\mathbf{t}_p = [t_{p1}, t_{p2}, \dots, t_{pM}]^T$, the sum of squared errors to be minimized is given by

$$E_p = \left(\frac{1}{2} \right) \sum_{j=1}^M t_{pj} - o_{pj}^2, \quad (13)$$

where M is the number of output nodes. By minimizing the error E_p using the gradient descent technique, the connection weights can be updated by applying the following equations:

$$\Delta w_{ji}(p) = \eta \delta_{pj} o_{pj} + \alpha \Delta w_{ji}(p-1), \quad (14)$$

where for output nodes,

$$\delta_{pj} = (t_{pj} - o_{pj}) o_{pj} (1 - o_{pj}), \quad (15)$$

and for hidden nodes,

$$\delta_{pj} = o_{pj} (1 - o_{pj}) \sum_k (\delta_{pk} \times w_{kj}). \quad (16)$$

Note that the learning rate affects the network's generalization ability and learning speed to a great extent.

2.3. Support Vector Regression. SVR is an adaptation of the support vector machine (SVM), one of the most powerful techniques in CI [24, 25]. The basic concept of SVR is to find a model function $f(x)$ to represent the relationship between the features and target. The modeling of SVR can be described as follows. Suppose

$$\mathbf{y} = f(\mathbf{x}) = (\mathbf{w} \cdot \Phi(\mathbf{x})) + b, \quad (17)$$

where \mathbf{x} and \mathbf{y} represent the model input and output vectors, \mathbf{w} is the weight vector, b is a constant, $\Phi(\mathbf{x})$ denotes a mapping function in the feature space, and $(\mathbf{w} \cdot \Phi(\mathbf{x}))$ describes the dot production in the feature space F .

Typical regression modeling estimates the coefficients by minimizing the square error, which can be considered as an empirical risk based on loss function. The ε -insensitivity loss function was introduced [25] and can be described as follows:

$$L_\varepsilon(f(x), y) = \begin{cases} |f(x) - y| - \varepsilon, & \text{if } |f(x) - y| \geq \varepsilon, \\ 0, & \text{otherwise,} \end{cases} \quad (18)$$

where y is the model output and ε is the region of ε -insensitivity. When the predicted value falls into the band area, the loss is zero. However, when the predicted value falls

outside the band area, the loss is defined as the difference between the predicted value and margin.

When both empirical risk and structure risk are considered, the SVR can be designed to minimize the following quadratic programming problem [25]:

$$\begin{aligned} \min \quad & \left(\frac{1}{2}\right)\|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*), \\ \text{subject to} \quad & (y_i - (w \cdot \Phi(x_i))) - b \leq \varepsilon + \xi_i, \\ & (w \cdot \Phi(x_i)) + b - y_i \leq \varepsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, \\ & \text{for } i = 1, \dots, n, \end{aligned} \quad (19)$$

where $i = 1, \dots, n$ is the number of training observations; ξ_i, ξ_i^* is the empirical risk; $(1/2)\|w\|^2$ is the structure risk preventing overlearning and lack of applied universality; and C is a modifying coefficient representing the trade-off between empirical and structure risks. With an appropriate modifying coefficient C , band area width ε , and kernel function, the optimum value of each parameter can be solved by the Lagrange procedure. In addition, the general form of the SVR-based regression function can be expressed as follows [25]:

$$f(x, w) = f(x, r, r^*) = \sum_{i=1}^N (r_i - r_i^*) K(x, x_i) + b, \quad (20)$$

where r_i and r_i^* are the Lagrangian multipliers and satisfy the equality $r_i r_i^* = 0$. In addition, $K(x, x_i)$ is the kernel function. The values of the kernel are equal to the inner product of two vectors, \mathbf{x}_i and \mathbf{x}_j , in the feature space $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$; that is, $K(x_i, x_j) = \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j)$. Because the radial basis function (RBF) is the most commonly used kernel function [26], we employ it for the experimental study. The RBF is written as follows:

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right), \quad (21)$$

where σ is the width of the RBF.

2.4. Multivariate Adaptive Regression Splines. MARS was developed for solving regression-type problems [27]. It is a nonparametric regression procedure that makes no assumption about the functional relationship between the response and explanatory variables. MARS modeling is based on a divide-and-conquer strategy where training datasets are partitioned into separate regions, each of which is assigned to its own regression equation. Consequently, MARS is appropriate and effective for problems with more than two input variables. Particularly, MARS can select important explanatory variables and relationships for complex data structures that often hide in higher dimensional data series.

A MARS model can be described as follows [27]:

$$f(x) = \beta_0 + \sum_{m=1}^M \beta_m \prod_{j=1}^{J_m} [S_{jm}(x_{v(j,m)} - l_{jm})], \quad (22)$$

where β_0 and β_m are the parameters, M is the number of basis functions (BFs), J_m is the number of knots, S_{jm} takes on values of either 1 or -1 and indicates the right or left sense of the associated step function, $v(j, m)$ is the label of the independent variable, and l_{jm} is the knot location. The optimal MARS model is determined in a two-step procedure. In the first step, a model is grown by adding BFs until an overly large model is obtained. The BFs are then deleted in the order of least contribution to most contribution by using the generalized cross-validation (GCV) criterion in the second step. The measure of variable importance is provided by observing the decrease in the computed GCV values when a variable is removed from the model. The GCV is described as follows:

$$\text{GCV}(M) = \frac{(1/N) \sum_{i=1}^N [y_i - f_M(x_i)]^2}{[1 - (C(M)/N)]^2}, \quad (23)$$

where N is the number of observations and $C(M)$ is the cost penalty measures of a model containing M BF.

3. Food Crop Price Forecasting

3.1. Datasets and Forecasting Criteria. In this study, the price data of the three most important food crops were collected for the period of January 1990 to September 2015 from the web sites of IndexMundi [28–31]. The datasets consist of 309 records for each food crop price. Among them, the first 297 data records were used to develop the different forecasting models, while the remaining 12 data records were used to perform model validation. This study has presented an experiment based on a larger population of data and has foreseen obtaining accurate forecasts of food crop price, with the forecast horizon for the out-of-sample forecasting experiment of 12. Good prediction of prices may assist in planning agricultural supply, but the investigated data does not contain any information on population.

The forecasting capability of the models was compared using three forecasting accuracy criteria, including mean absolute percentage error (MAPE), root mean square error (RMSE), and mean absolute error (MAE). These forecasting measurements are expressed as follows:

$$\begin{aligned} \text{MAPE} &= \left(\frac{1}{n}\right) \sum_{t=1}^n \frac{|e_t|}{Y_t} \times 100\%, \\ \text{MAE} &= \left(\frac{1}{n}\right) \sum_{t=1}^n |e_t|, \\ \text{RMSE} &= \sqrt{\left(\frac{1}{n}\right) \sum_{t=1}^n (e_t)^2}, \end{aligned} \quad (24)$$

where e_t is the value of the residual at time t and Y_t is the observation at time t .

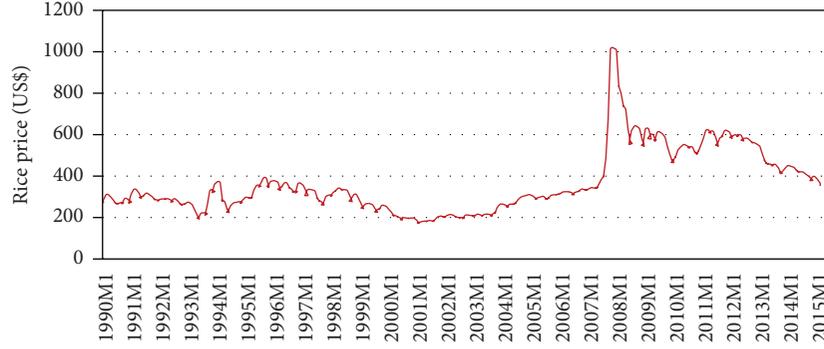


FIGURE 1: Time plot for rice price (unit: US dollars per metric ton).

Obviously, it can be noted that the lower the MAPE, M AE, and RMSE values, the closer the forecasted values to the actual values.

3.2. Forecast Modeling of Rice Prices. Figure 1 shows the original time plot for the rice price data series. This study uses an SAS package to run the ARIMA modeling. Table 1 shows the parameter estimates, and all the estimated parameters are significantly different from zero. In Table 1, the notations “AR1,1,” “AR1,2,” “AR1,3,” and “AR1,4” correspond to the parameters ϕ_1 , ϕ_2 , ϕ_7 , and ϕ_{13} of the AR model. The ARIMA model presented in Table 1 was a subset model. Since the parameters, ϕ_1 , ϕ_2 , ϕ_7 , and ϕ_{13} , are significantly different from zero, this study refers to the model in Table 1 as an AR (1, 2, 7, and 13) model.

In addition, Table 2 shows the Ljung-Box test results. As mentioned earlier, the Ljung-Box statistic can be used to check the fit of an ARIMA model. A simple method to derive the hypothesis testing result is described as follows. We should reject the model under consideration by setting the type I error equal to α if and only if the p value is less than α . Generally, α is set equal to 0.05. The SAS package computes the Ljung-Box statistics Q and their associated p values for values of K equal to 6, 12, 18, 24, 30, and 36. In Table 2, the first column contains the values of K , the second column contains the values of Q , and the fourth column contains the associated p values. By studying Table 2, we observe that all the associated p values are greater than α . Accordingly, a conclusion can be made concerning the fit of the underlying ARIMA model. Thus, the model expressed in (25) is suitable for modeling rice prices.

$$Z_t = 0.46514Z_{t-1} - 0.502Z_{t-2} - 0.17055Z_{t-7} - 0.11264Z_{t-13} + a_t, \quad (25)$$

where

$$Z_t = (y_t - y_{t-1}). \quad (26)$$

For ANN designs, there is no fixed mode to decide the number of hidden nodes. Too few hidden nodes confine the network generalization capability, whereas too many hidden nodes may lead to overtraining difficulties. Therefore, in this study, we consider the hidden nodes to be set from $(2n - 2)$ to $(2n + 2)$ if $n \leq 8$, where n represents the number of input

TABLE 1: Parameter estimates for rice prices.

Parameter	The ARIMA procedure				Lag
	Estimate	Standard error	t value	Approx Pr > t	
AR1,1	0.46514	0.05586	8.33	<0.0001	1
AR1,2	-0.23278	0.05559	-4.19	<0.0001	2
AR1,3	-0.17055	0.05183	-3.29	0.0011	7
AR1,4	-0.11264	0.05170	-2.18	0.0301	13

variables. When $n > 8$, we consider the hidden nodes to be set from $(n - 2)$ to $(n + 2)$. In this study, we denote the term $\{n_i - n_h - n_o\}$ for ANN parameter settings, where n_i is the number of neurons in the input layer, n_h stands for the number of neurons in the hidden layer, and n_o represents the number of neurons in the output layer, respectively. Additionally, this study employs the learning rate for all ANN models at the default value (i.e., 0.01) to ensure consistency [32]. The network topology with the minimum MAPE is considered as the optimal network topology, due to the fact that the MAPE is one of the most important performance measurements of forecasting capability.

In this study, because there are no explanatory variables available, the inputs of ANN modeling for food crop price forecasting becomes unfeasible. Consequently, this study uses self-predictor variables to serve as the inputs for ANN. In here, the self-predictor variable is defined as the lags or past values of a variable. This study proposes two input designs for the modeling of CI techniques. Because seasonal effects may influence food crop price forecasts, the first design rationally selects the preceding 12 observations to forecast food crop prices at time t . Accordingly, the first design used 12 self-predictor variables (i.e., y_{t-1} , y_{t-2} , y_{t-3} , ..., and y_{t-12}) to serve as input variables and employed a single variable (i.e., y_t) to serve as the output for ANN modeling. That is, the first design considered 12 input nodes and one output node for the ANN structures. This design is denoted as ANN₁.

The second input design for CI technique modeling is performed using FSM. For this proposed modeling, this study incorporated FSMs with ANN, SVR, and MARS to develop forecasting models. The proposed input design used ARIMA as an FSM to extract important self-predictor

TABLE 2: Ljung-Box test results for ARIMA modeling of rice prices.

To Lag	Chi-square	DF	Autocorrelation check of residuals						
			Pr > ChiSq	Autocorrelations					
6	4.61	2	0.0995	-0.002	0.007	-0.024	-0.073	0.006	-0.096
12	6.34	8	0.6097	0.009	0.007	0.016	0.027	-0.031	0.059
18	13.10	14	0.5190	-0.023	0.028	0.028	-0.09	0.061	-0.087
24	14.40	20	0.8096	0.059	0.012	0.008	-0.006	-0.012	-0.015
30	19.29	26	0.8242	-0.102	-0.045	-0.025	-0.002	0.026	0.036
36	20.10	32	0.9494	0.018	-0.022	-0.002	-0.024	-0.003	0.032

TABLE 3: Forecasting measurements for two ANN designs (rice price forecasting).

ANN ₁ topology	MAE	RMSE	MAPE (%)	ARIMA-ANN topology	MAE	RMSE	MAPE (%)
{12-10-1}	15.438	16.648	3.939	{4-6-1}	10.384	12.053	2.654
{12-11-1}	14.737	16.022	3.765	{4-7-1}	10.365	12.037	2.650
{12-12-1}	15.001	16.262	3.825	{4-8-1}	10.374	12.044	2.652
{12-13-1}	15.182	16.499	3.869	{4-9-1}	10.379	12.062	2.653
{12-14-1}	13.915	15.320	3.557	{4-10-1}	10.389	12.039	2.655

variables, which serve as the inputs for the ANN, SVR, and MARS models. Because the ARIMA model for rice price forecasting is obtained in (25), it indicates that Z_{t-1} , Z_{t-2} , Z_{t-7} , and Z_{t-13} will influence Z_t . Consequently, this study selected Z_{t-1} , Z_{t-2} , Z_{t-7} , and Z_{t-13} to serve as self-predictor, or input, variables for the proposed integrated modeling on the rice price data series. In this study, this second input design is denoted as ARIMA-ANN.

To forecast the rice prices, the two ANN models contain twelve and four input nodes for the first and second designs, respectively. The hidden nodes were selected as 10, 11, 12, 13, and 14 for the first design and 6, 7, 8, 9, and 10 for the second design. After performing the ANN modeling, the ANN₁ design showed that the {12-14-1} structure provided the best results and minimum testing MAPE for rice prices. For the ARIMA-ANN design, the best structure was {4-7-1}. Three forecasting measurements for different settings of the ANN topologies for the two designs are shown in Table 3.

For modeling SVR on rice prices, this study adopted the input structure as ANN modeling. SVR modeling employed two designs for the input variables. The first design used 12 variables, y_{t-1} , y_{t-2} , y_{t-3} , \dots , and y_{t-12} as the input variables and y_t as the output variable. The second design used four self-predictor variables, Z_{t-1} , Z_{t-2} , Z_{t-7} , and Z_{t-13} to serve as the input variables and Z_t as the output variable. The first and second SVR designs are denoted as SVR₁ and ARIMA-SVR, respectively. Because the parameter settings of C and ϵ often affect the performance of SVR modeling, an analytic parameter selection approach and grid search are used in this study [26]. Accordingly, we denote the term $\{\epsilon, C\}_{\text{SVR}}$ as the best SVR parameter settings. After performing the SVR modeling, the SVR₁ design reported that the parameter settings of $\{2^{-4}, 2^{11}\}_{\text{SVR}}$ provided MAPE=6.912% for rice price forecasting. For the second design, ARIMA-SVR, the parameter settings of $\{2^{-5}, 2^9\}_{\text{SVR}}$ provided MAPE=2.057%.

For MARS modeling, this study also adopted the same input structure of ANN or SVR modeling. The first and second MARS designs are denoted as MARS₁ and ARIMA-MARS, respectively. Table 4 lists the variable selection results and BFs after modeling rice price data using MARS₁. In Table 4, the first column contains the variables that should be included in the model, the second column contains the relative importance of the variables that are listed in the first column, the third column contains the various BFs, and the final column contains the estimated coefficient values for the BFs that are listed in the third column. Based on the results in Table 4, it can be seen that three input variables (i.e., y_{t-12} , y_{t-11} , and y_{t-10}) played important roles in building the MARS forecasting models. Their relative importance values (%) are 100, 13.1, and 12.1. The construction of the BFs can be expressed as follows. Consider the variable y_{t-12} as an example. We have two BFs (i.e., BF₁ and BF₂) to be considered. The observed values of BF₁ are determined by the values larger than 0 or $y_{t-12} - 367$. The corresponding coefficient is estimated to be 0.985. The observed values of BF₂ are determined by the values larger than 0 or $552.09 - y_{t-12}$. The corresponding coefficient is estimated to be -1.258. Accordingly, the MARS forecasting model for rice prices can be expressed as follows:

$$\begin{aligned}
Y = & 542.484 + 0.985(\text{BF}_1) - 1.258(\text{BF}_2) \\
& + 1.539\text{BF}_3 - 1.880(\text{BF}_4) + 0.524(\text{BF}_5) \\
& - 2.410(\text{BF}_6) + 1.783(\text{BF}_7) + 0.454(\text{BF}_8) \\
& - 0.234(\text{BF}_9) + 0.346(\text{BF}_{10}).
\end{aligned} \tag{27}$$

For ARIMA-MARS modeling, the variable selection results and BFs are summarized in Table 5. In addition, it can be seen that three input variables (i.e., Z_{t-13} , Z_{t-7} , and Z_{t-1}) played important roles in building the ARIMA-MARS

TABLE 4: The results of MARS₁ for rice price forecasting.

Variable selection results			
Variable	Relative importance (%)	BF	Coefficient
Intercept			542.484
y_{t-12}	100.0	$BF_1 = \text{Max}$ (0, $y_{t-12} - 367$)	0.985
		$BF_2 = \text{Max}$ (0, $552.09 - y_{t-12}$)	-1.258
		$BF_3 = \text{Max}$ (0, $y_{t-11} - 356$)	1.539
y_{t-11}	13.1	$BF_4 = \text{Max}$ (0, $y_{t-11} - 472.48$)	-1.880
		$BF_5 = \text{Max}$ (0, $585.95 - y_{t-11}$)	0.524
		$BF_6 = \text{Max}$ (0, $y_{t-10} - 363$)	-2.410
y_{t-10}	12.1	$BF_7 = \text{Max}$ (0, $y_{t-10} - 472.48$)	1.783
		$BF_8 = \text{Max}$ (0, $y_{t-10} - 543.14$)	0.454
		$BF_9 = \text{Max}$ (0, $623 - y_{t-10}$)	-0.234
		$BF_{10} = \text{Max}$ (0, $y_{t-10} - 623$)	0.346

forecasting models. The MARS forecasting model for rice prices can be described as follows:

$$Y = -13.404 + 0.407(BF_1) + 23.543(BF_2) - 73.468(BF_3) + 49.716(BF_4) + 0.479(BF_5), \quad (28)$$

where

$$Z_t = (y_t - y_{t-1}). \quad (29)$$

3.3. Forecast Modeling of Wheat Prices. Figure 2 displays the original time plot for wheat prices. After modeling the wheat price data series with the ARIMA procedure, the parameter estimates are shown in Table 6. In Table 6, the notations “MA1,1,” “MA1,2,” “MA1,3,” and “MA1,4” correspond to the parameters θ_1 , θ_7 , θ_{12} , and θ_{13} of the MA model. Table 7 displays the Ljung-Box test results, and a conclusion is made on the appropriateness of the underlying ARIMA model. Accordingly, (30) is suitable for modeling wheat prices.

$$Z_t = a_t - 0.2402a_{t-1} + 0.14629a_{t-7} + 0.11171a_{t-12} + 0.13721a_{t-13}, \quad (30)$$

where

$$Z_t = y_t - y_{t-1}. \quad (31)$$

In addition, a_{t-j} stands for the white noise at time $t - j$.

For ANN modeling on wheat prices, this study also used two ANN designs, same as the structures for modeling rice prices. The first design used 12 self-predictor variables (i.e.,

TABLE 5: The results of ARIMA-MARS for rice price forecasting.

Variable selection results			
Variable	Relative importance (%)	BF	Coefficient
Intercept			-13.404
Z_{t-13}	100.0	$BF_1 = \text{Max}$ (0, $Z_{t-13} + 35$)	0.407
		$BF_2 = \text{Max}$ (0, $Z_{t-7} - 0$)	23.543
Z_{t-7}	45.4	$BF_3 = \text{Max}$ (0, $Z_{t-7} + 1.30$)	-73.468
		$BF_4 = \text{Max}$ (0, $Z_{t-7} + 1.87$)	49.716
Z_{t-1}	28.5	$BF_5 = \text{Max}$ (0, $-39.85 - Z_{t-1}$)	0.479

y_{t-1} , y_{t-2} , y_{t-3} , ..., and y_{t-12}) to serve as input variables and a single variable (i.e., y_t) to serve as the output variable. The second design considers the ARIMA as an FSM to extract important self-predictor variables, which serve as the inputs for the ANN models. The ARIMA model for wheat price forecasting is obtained in (30) and indicates that a_{t-1} , a_{t-7} , a_{t-12} , and a_{t-13} will influence Z_t .

Because $a_{t-1} = Z_{t-1} - Z_{t-1}^{\wedge}$, where Z_{t-1}^{\wedge} is the forecast at time $t - 1$, we observe that Z_{t-1} will influence Z_t . Therefore, Z_{t-1} should be selected as an input variable as long as the component a_{t-1} is in the MA model. Following the same logic, because the model (e.i., (30)) contains a_{t-1} , a_{t-7} , a_{t-12} , and a_{t-13} , we conclude that Z_{t-1} , Z_{t-7} , Z_{t-12} , and Z_{t-13} will influence Z_t . Accordingly, we have selected Z_{t-1} , Z_{t-7} , Z_{t-12} , and Z_{t-13} to serve as input variables for the second proposed ANN modeling design for the wheat price data series.

After performing ANN modeling on wheat price data, Table 8 shows the corresponding forecast validity measures for different settings of ANN topologies for the two designs. As shown in Table 8, we observe that the ANN₁ design with the {12-11-1} structure has the smallest MAPE. For the second, or ARIMA-ANN design, the {4-8-1} structure was associated with the smallest MAPE.

For modeling SVR on wheat prices, this study adopted the input structure as ANN modeling. The first design employed 12 variables, y_{t-1} , y_{t-2} , y_{t-3} , ..., and y_{t-12} as the input variables and y_t as the output variable. The second design employed four self-predictor variables, Z_{t-1} , Z_{t-7} , Z_{t-12} , and Z_{t-13} to serve as the input variables and Z_t as the output variable. After performing SVR modeling on the wheat price data series, we obtained the parameter settings of $\{2^{-8}, 2^{10}\}_{\text{SVR}}$ and associated MAPE = 4.661% for the first design. In addition, we had parameter settings of $\{2^{-7}, 2^{-1}\}_{\text{SVR}}$ and associated MAPE = 4.347% for the second design.

For MARS modeling with the first design, Table 9 presents the variable selection results and BFs. As shown in Table 9, we notice that three input variables (i.e., y_{t-12} , y_{t-6} , and y_{t-11}) played important roles in building the MARS forecasting models. In addition, the MARS forecasting model for wheat prices can be expressed as follows:

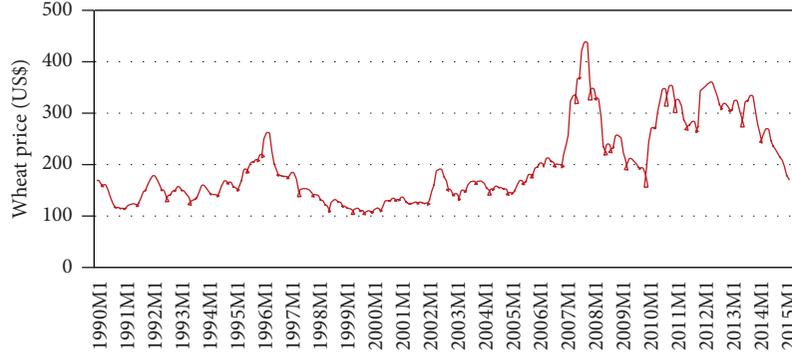


FIGURE 2: Time plot for wheat price (unit: US dollars per metric ton).

TABLE 6: Parameter estimates for wheat prices.

The ARIMA procedure					
Conditional least squares estimation					
Parameter	Estimate	Standard error	t value	Approx Pr > $ t $	Lag
MA1,1	-0.2402	0.05601	-4.29	<0.0001	1
MA1,2	0.14629	0.05754	2.54	0.0115	7
MA1,3	0.11171	0.0599	1.86	0.0632	12
MA1,4	0.13721	0.06018	2.28	0.0233	13

$$Y = 360.973 + 0.765(\text{BF}_1) - 1.272(\text{BF}_2) - 0.185(\text{BF}_3) + 0.298(\text{BF}_4) - 0.218(\text{BF}_5). \quad (32)$$

For MARS modeling with the second design, the variable selection results and BFs are summarized in Table 10. Furthermore, the MARS forecasting model for wheat prices is expressed as follows:

$$Y = -13.665 - 1.105(\text{BF}_1) + 8.333(\text{BF}_2) - 13.064(\text{BF}_3) + 4.955(\text{BF}_4) - 0.155(\text{BF}_5). \quad (33)$$

3.4. Forecast Modeling of Corn Prices. Figure 3 presents the original time plot for corn prices. Table 11 presents the parameter estimates after performing ARIMA modeling. In Table 11, the notations “AR1,1,” “AR1,2,” “AR1,3,” “AR1,4,” “AR1,5,” and “AR1,6” correspond to as parameters ϕ_1 , ϕ_3 , ϕ_6 , ϕ_{12} , ϕ_{14} , and ϕ_{24} of the AR model. In Table 11, we still contain the parameters of “AR1,2” and “AR1,4” in the model, although the absolute t values are less than or equal to 2 (i.e., the typical type I error is chosen as 0.05). The main reason is that those two parameters are important, and the Ljung-Box statistics indicate that the model is not appropriate if those two parameters are not involved in the underlying ARIMA model. Additionally, Table 12 demonstrates the Ljung-Box test results and a conclusion is made on the appropriateness of the ARIMA model. Thus, (34) is a suitable ARIMA model for modeling the corn price data series.

$$Z_t = 0.17626Z_{t-1} + 0.09344Z_{t-3} - 0.145Z_{t-6} - 0.0925Z_{t-12} - 0.1476Z_{t-14} - 0.2149Z_{t-24} + a_t, \quad (34)$$

where

$$Z_t = (y_t - y_{t-1}). \quad (35)$$

For ANN modeling on corn prices, while the first design used 12 input variables, the second design used six input variables extracted by using the FSM of ARIMA modeling. After performing ANN modeling, Table 13 presents the corresponding forecast validity measures for different ANN topology settings for the two designs. From Table 13, we observe that the ANN₁ design with the {12-14-1} structure has the smallest MAPE. For the second design, the {6-13-1} structure has the smallest MAPE.

In the modeling of SVR for the corn price data series, this study adopted the same input structure as ANN modeling. The first design employed 12 variables (i.e., y_{t-1} , y_{t-2} , y_{t-3} , ..., and y_{t-12}) as the input variables and y_t as the output variable. The second design employed six self-predictor variables (i.e., Z_{t-1} , Z_{t-3} , Z_{t-6} , Z_{t-12} , Z_{t-14} , and Z_{t-24}), which were selected by using the FSM of ARIMA modeling to serve as the input variables and Z_t as the output variable. After performing SVR modeling, we obtained the parameter settings of $\{2^{-9}, 2^7\}_{\text{SVR}}$ and associated MAPE = 4.060% for the first design. In addition, we obtained parameter settings of $\{2^{-11}, 2^{11}\}_{\text{SVR}}$ and associated MAPE = 4.215% for the second design.

For MARS modeling on corn price data with the first design, Table 14 illustrates the variable selection results and BFs. As shown in Table 14, we notice that two variables (i.e., y_{t-12} and y_{t-11}) play important roles in building the MARS forecasting models. Consequently, the MARS forecasting model for corn prices can be described as follows:

$$Y = 262.148 - 1.354(\text{BF}_1) + 0.969(\text{BF}_2) + 0.342(\text{BF}_3). \quad (36)$$

For MARS modeling with the second design, Table 15 lists the variable selection results and BFs. The MARS forecasting model for the corn prices is expressed as follows:

TABLE 7: Ljung-Box test results for ARIMA modeling of wheat prices.

To Lag	Chi-square	DF	Autocorrelation check of residuals						
			Pr > ChiSq	Autocorrelations					
6	4.4	2	0.111	0.019	-0.045	0.024	-0.036	0.069	0.074
12	11.07	8	0.1977	-0.03	-0.126	0.008	-0.055	0.042	-0.001
18	14.41	14	0.4197	-0.035	-0.084	0.021	-0.038	0.021	0.006
24	19.99	20	0.4589	0.067	0.081	-0.033	-0.03	0.012	-0.064
30	30.52	26	0.2465	-0.007	-0.041	-0.11	-0.132	-0.009	0.026
36	33.96	32	0.3733	-0.025	0.025	-0.071	0.011	0.054	0.029

TABLE 8: Forecast validity measures for two ANN designs (rice price forecasting).

ANN ₁ topology	MAE	RMSE	MAPE (%)	ARIMA-ANN topology	MAE	RMSE	MAPE (%)
{12-10-1}	12.121	13.521	5.448	{4-6-1}	11.833	14.336	5.374
{12-11-1}	11.752	13.230	5.285	{4-7-1}	11.865	14.353	5.388
{12-12-1}	12.084	13.515	5.423	{4-8-1}	8.984	10.864	4.077
{12-13-1}	11.984	13.429	5.385	{4-9-1}	8.984	10.883	4.078
{12-14-1}	12.120	13.516	5.457	{4-10-1}	11.845	14.346	5.379

TABLE 9: The results of MARS₁ for wheat price forecasting.

Variable selection results			
Variable	Relative importance (%)	BF	Coefficient
Intercept			360.973
y_{t-12}	100.0	$BF_1 = \text{Max}$ (0, $y_{t-12} - 321.18$)	0.765
		$BF_2 = \text{Max}$ (0, $321.81 - y_{t-12}$)	-1.272
		$BF_3 = \text{Max}$ (0, $y_{t-6} - 178.32$)	-0.185
y_{t-6}	7.2	$BF_4 = \text{Max}$ (0, $178.32 - y_{t-6}$)	0.298
		$BF_5 = \text{Max}$ (0, $y_{t-11} - 269.03$)	-0.218

$$Y = 3.787 - 0.277(BF_1) - 0.745(BF_2) - 0.466(BF_3) + 0.175(BF_4). \quad (37)$$

4. Forecasting Comparison

Several forecasting models were proposed to forecast the three major food crop prices in this study. These models include four single models (i.e., ARIMA, ANN₁, SVR₁, and MARS₁) and three integrated models with FSMs (i.e., ARIMA-ANN, ARIMA-SVR, and ARIMA-MARS). Tables 16–18 present the forecasting results, as well as the MAE, RMSE, and MAPE values of the forecasting models for rice, wheat, and corn price forecasting, respectively. Low MAE, RMSE, or MAPE values are associated with better forecasting accuracy.

In comparison to the forecasting performance of the first design, or single models, in Tables 16–18, we observe that the three CI models demonstrated better performance than the ARIMA models. The possible reason may be that ARIMA

TABLE 10: The results of ARIMA-MARS for wheat price forecasting.

Variable selection results			
Variable	Relative importance (%)	BF	Coefficient
Intercept			-13.665
Z_{t-7}	100.0	$BF_1 = \text{Max}$ (0, $Z_{t-7} - 26.58$)	-1.105
		$BF_2 = \text{Max}$ (0, $Z_{t-13} + 26.91$)	8.333
Z_{t-13}	63.3	$BF_3 = \text{Max}$ (0, $Z_{t-13} + 22.55$)	-13.064
		$BF_4 = \text{Max}$ (0, $Z_{t-13} + 17.56$)	4.955
Z_{t-1}	14.4	$BF_5 = \text{Max}$ (0, $Z_{t-1} + 10.65$)	-0.155

modeling is difficult for capturing nonlinear features in the food crop price data series. By reviewing these tables, we found that there is no best model for food crop price forecasting. For example, while the ANN₁ model seems to possess better forecasting accuracy for rice price forecasting, the SVR₁ model seems to have better forecasting accuracy for wheat and corn price forecasting.

It can be clearly seen in Tables 16–18 that the proposed integrated models possess better forecasting accuracy than the single models for most cases. For example, by reviewing Table 16, the proposed integrated ARIMA-ANN model is associated with MAE, RMSE, and MAPE values of 10.365%, 12.037%, and 2.650%, respectively, for rice price forecasting. These three performance values are smaller than the corresponding performance values for any one of the four single models. Take the proposed integrated ARIMA-SVR model as another example. By reviewing Table 17, the ARIMA-SVR model is associated with MAE, RMSE, and MAPE values of 9.779%, 11.371%, and 4.347%, respectively, for wheat price forecasting. These three performance values

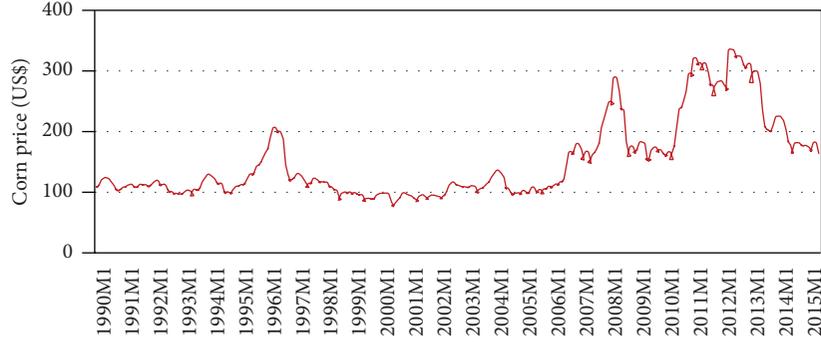


FIGURE 3: Time plot for corn price (unit: US dollars per metric ton).

TABLE 11: Parameter estimates for corn prices.

Parameter	The ARIMA procedure Conditional least squares estimation				Lag
	Estimate	Standard error	t value	Approx Pr > $ t $	
AR1,1	0.17626	0.05909	2.98	0.0031	1
AR1,2	0.09344	0.0651	1.44	0.1524	3
AR1,3	-0.145	0.06527	-2.22	0.0272	6
AR1,4	-0.0925	0.06619	-1.4	0.1636	12
AR1,5	-0.1476	0.06719	-2.2	0.029	14
AR1,6	-0.2149	0.07024	-3.06	0.0024	24

are also smaller than the corresponding performance values for any one of the four single models in Table 17. Thus, our proposed models, which were integrated with FSM, provide more accurate forecasting results than the single models.

In addition, Table 19 presents a comparison with respect to the overall percentage improvements (PIs) of forecasting accuracy for the proposed integrated models over the single models. The PIs of the MAE, RMSE, and MAPE are defined as follows:

$$\begin{aligned}
 \text{MAE}_{\text{PI}} &= \frac{[\text{MAE}(\text{single model}) - \text{MAE}(\text{integrated model})]}{\text{MAE}(\text{integrated model})} * 100\%, \\
 \text{RMSE}_{\text{PI}} &= \frac{[\text{RMSE}(\text{single model}) - \text{RMSE}(\text{integrated model})]}{\text{RMSE}(\text{integrated model})} * 100\%, \\
 \text{MAPE}_{\text{PI}} &= \frac{[\text{MAPE}(\text{single model}) - \text{MAPE}(\text{integrated model})]}{\text{MAPE}(\text{integrated model})} * 100\%.
 \end{aligned}
 \tag{38}$$

From Table 19, it is obvious that positive PIs can be achieved by using the proposed integrated models. For example, in the rice price forecasting, the MAPE_{PI} of the proposed ARIMA-ANN model over the two single models, ARIMA and ANN_{S} , were 376.038% and 34.226%, respectively. The corresponding MAE_{PI} and RMSE_{PI} were 369.822% and 34.250% and 340.945% and 27.273%, respectively. Apart from the rice price forecasting, most of the MAE_{PI} , RMSE_{PI} , and MAPE_{PI} are positive large numbers for wheat and corn price forecasting. Accordingly, considerable forecasting accuracy improvements are achieved through the use

of the proposed integrated models. Furthermore, we also found five negative PIs in Table 19.

For wheat price forecasting, the RMSE_{PI} of the ARIMA-MARS over MARS_1 was -1.029%. For corn price forecasting, the RMSE_{PI} of the ARIMA-ANN over ANN_1 was -10.973%. The three remaining negative values occurred in corn price forecasting when using the ARIMA-SVR model, and the MAE_{PI} , RMSE_{PI} , and MAPE_{PI} were -2.601%, -5.525%, and -3.677%, respectively. A negative PI value implies that the forecasting performance declined. However, the magnitudes of those five negative values are small, and they have minor effects on forecasting performance. Additionally, the associated MAPE_{PI} for those five negative PIs were only one negative. That is, if a forecasting model has the minimum MAPE, this does not mean it has the minimum MAE or RMSE.

5. Discussion

While the aforementioned description focuses on accuracy comparisons for each individual food crop price forecast, the following discussion provides the comparison regarding overall performance for food crop price forecasting as a whole.

Table 20 lists the average PIs of the proposed integrated models over their common element, the ARIMA models. Figure 4 presents the average PIs of MAE, RMSE, and MAPE by employing the proposed integrated models over single ARIMA models. As shown in Figure 4, we notice that considerable accuracy improvements can be reached by using our proposed approaches. Additionally, Table 21 reports the average PIs of MAE, RMSE, and MAPE by using the proposed integrated models over another element, other than ARIMA element. Figure 5 displays the satisfied average PIs by using the proposed integrated models.

In addition, Figure 6 displays the forecasts of rice prices, obtained by using ARIMA, ANN_1 , and the proposed ARIMA-ANN models, for the twelve testing records. One can see that the forecasts of the ARIMA-ANN model are closest to the actual observations. The forecasts of the basic ARIMA model are relatively far away from the actual observations. Additionally, the prediction accuracy of ANN_1 is better than that of the ARIMA model. Regarding the ANN_1 design, we observe that the input vectors $\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_{12}$ are equivalent to $y_{t-1}, y_{t-2}, y_{t-3}, \dots, y_{t-12}$ and that the output O

TABLE 12: Ljung-Box test results for ARIMA modeling of corn prices.

To Lag	Chi-square	DF	Autocorrelation check of residuals						
			Pr > ChiSq	Autocorrelations					
6	.	0	.	-0.011	0.082	0.026	0.002	-0.063	-0.027
12	6.71	6	0.3481	0.013	-0.032	0.001	0.012	0.091	-0.014
18	14.3	12	0.2822	-0.069	-0.036	0.042	0.113	0.038	-0.046
24	16.43	18	0.5624	0.014	-0.033	-0.015	0.061	-0.034	-0.015
30	29.11	24	0.2159	0.035	-0.073	0.103	-0.079	-0.025	-0.12
36	30.86	30	0.4225	-0.036	0	0.06	0.005	0.015	-0.001

TABLE 13: Forecast validity measures for two ANN designs (corn price forecasting).

ANN ₁ topology	MAE	RMSE	MAPE (%)	ARIMA-ANN topology	MAE	RMSE	MAPE (%)
{12-10-1}	7.536	9.698	4.387	{6-10-1}	7.304	10.773	4.267
{12-11-1}	7.464	9.635	4.343	{6-11-1}	7.307	10.853	4.269
{12-12-1}	7.444	9.638	4.331	{6-12-1}	7.291	10.805	4.259
{12-13-1}	7.487	9.633	4.357	{6-13-1}	7.286	10.771	4.256
{12-14-1}	7.433	9.589	4.326	{6-14-1}	7.290	10.809	4.259

TABLE 14: The results of MARS₁ for corn price forecasting.

Variable selection results			
Variable	Relative importance (%)	BF	Coefficient
Intercept			262.148
y_{t-12}	100.0	BF ₁ = Max (0, 268.79 - y_{t-12})	-1.354
		BF ₂ = Max (0, y_{t-12} - 268.79)	0.969
y_{t-11}	7.2	BF ₃ = Max (0, 295.29 - y_{t-11})	0.342

TABLE 15: The results of ARIMA-MARS for corn price forecasting.

Variable selection results			
Variable	Relative importance (%)	BF	Coefficient
Intercept			3.787
Z_{t-24}	100.0	BF ₁ = Max (0, 15.08 - Z_{t-24})	-0.277
Z_{t-12}	74.0	BF ₂ = Max (0, Z_{t-12} - 13.9)	-0.745
Z_{t-1}	44.8	BF ₃ = Max (0, Z_{t-1} - 11.59)	-0.466
Z_{t-6}	17.7	BF ₄ = Max (0, Z_{t-6} + 7.36)	0.175

can be obtained by performing a nonlinear functional mapping, as shown in (10). Regarding the ARIMA-ANN mechanism, we observe that the input vectors I_1, I_2, I_3 , and I_4 are characterized by $Z_{t-1}, Z_{t-2}, Z_{t-7}$, and Z_{t-13} and that the output O can be obtained by performing a nonlinear functional mapping, as shown in (10).

Figure 7 presents the forecasts of rice prices, obtained by using ARIMA, SVR₁, and the proposed ARIMA-SVR

models, for the twelve testing records. Regarding the SVR₁ design, we observe that the input vectors I_1, I_2, \dots, I_{12} are characterized by $y_{t-1}, y_{t-2}, y_{t-3}, \dots, y_{t-12}$ and that the output O can be obtained by performing a nonlinear functional mapping, as shown in (17). Regarding the ARIMA-SVR mechanism, we observe that the input vectors I_1, I_2, I_3 , and I_4 are characterized by $Z_{t-1}, Z_{t-2}, Z_{t-7}$, and Z_{t-13} and that the output O can be obtained by performing a nonlinear functional mapping, as shown in (17).

Figure 8 shows the forecasts of rice prices, obtained by using ARIMA, MARS₁, and the proposed ARIMA-MARS models, for the twelve testing records. Regarding MARS₁ modeling, we observe that the input vectors I_1, I_2, \dots, I_{12} are characterized by $y_{t-1}, y_{t-2}, y_{t-3}, \dots, y_{t-12}$ and that the output O can be obtained by performing a nonlinear functional mapping, as shown in (22). Regarding the ARIMA-MARS design, we observe that the input vectors I_1, I_2, I_3 , and I_4 are characterized by $Z_{t-1}, Z_{t-2}, Z_{t-7}$, and Z_{t-13} and that the output O can be obtained by performing a nonlinear functional mapping, as shown in (22). By observing Figures 7 and 8, we can see that the forecasts of the ARIMA-SVR and ARIMA-MARS models are closest to the actual observations. Both figures illustrate that the forecasts of the ARIMA models are relatively far away from the actual observations.

Similar implications were obtained for the cases of wheat and corn price predictions. That is, the forecasts of the proposed integrated models are closer to the actual observations. The forecasts of the ARIMA model are far away from the actual observations. These findings can be observed in Figures 9–14.

In this study, we collected 309 records to build various models for the prediction of the prices of rice, wheat, and corn. One assumption for building the ARIMA models is that their structures will remain unchanged over time. Therefore, even if more sample data become available, there is no need to rebuild the ARIMA model for our proposed integrated

TABLE 16: Performance comparison of various models for rice price forecasting.

	Inputs	MAE (%)	RMSE (%)	MAPE (%)
(1) Single models				
ARIMA		48.697	53.077	12.615
ANN ₁	$y_{t-1} \sim y_{t-12}$	13.915	15.320	3.557
SVR ₁	$y_{t-1} \sim y_{t-12}$	27.994	34.514	6.912
MARS ₁	$y_{t-10}, y_{t-11}, y_{t-12}$	20.392	25.090	5.196
(2) Integrated models				
ARIMA-ANN	$Z_{t-1}, Z_{t-2}, Z_{t-7}, Z_{t-13}$	10.365	12.037	2.650
ARIMA-SVR	$Z_{t-1}, Z_{t-2}, Z_{t-7}, Z_{t-13}$	7.958	9.786	2.057
ARIMA-MARS	$Z_{t-1}, Z_{t-7}, Z_{t-13}$	11.244	15.566	2.916

TABLE 17: Performance comparison of various models for wheat price forecasting.

	Inputs	MAE (%)	RMSE (%)	MAPE (%)
(1) Single models				
ARIMA		33.520	42.250	16.651
ANN ₁	$y_{t-1} \sim y_{t-12}$	11.752	13.230	5.285
SVR ₁	$y_{t-1} \sim y_{t-12}$	10.203	13.586	4.661
MARS ₁	$y_{t-6}, y_{t-11}, y_{t-12}$	12.804	13.794	5.667
(2) Integrated models				
ARIMA-ANN	$Z_{t-1}, Z_{t-7}, Z_{t-12}, Z_{t-13}$	8.984	10.883	4.078
ARIMA-SVR	$Z_{t-1}, Z_{t-7}, Z_{t-12}, Z_{t-13}$	9.779	11.371	4.347
ARIMA-MARS	$Z_{t-1}, Z_{t-7}, Z_{t-13}$	10.643	13.937	4.809

TABLE 18: Performance comparison of various models for corn price forecasting.

	Inputs	MAE (%)	RMSE (%)	MAPE (%)
(1) Single models				
ARIMA		14.783	20.047	8.784
ANN ₁	$y_{t-1} \sim y_{t-12}$	7.433	9.589	4.326
SVR ₁	$y_{t-1} \sim y_{t-12}$	7.003	9.058	4.060
MARS ₁	y_{t-11}, y_{t-12}	7.015	9.337	4.110
(2) Integrated models				
ARIMA-ANN	$Z_{t-1}, Z_{t-3}, Z_{t-6}, Z_{t-12}, Z_{t-14}, Z_{t-24}$	7.286	10.771	4.256
ARIMA-SVR	$Z_{t-1}, Z_{t-3}, Z_{t-6}, Z_{t-12}, Z_{t-14}, Z_{t-24}$	7.190	9.588	4.215
ARIMA-MARS	$Z_{t-1}, Z_{t-6}, Z_{t-12}, Z_{t-24}$	6.495	9.102	3.785

models. This is a significant advantage over existing models. In other words, the ARIMA feature selection procedure only needs to be performed once. Even if more sample data becomes available, we can still use the same input variables for our proposed ARIMA-ANN, ARIMA-SVR, and ARIMA-MARS models. Some feature selection modeling processes may need to be performed repeatedly when more sample data become available, which may be a time-consuming task. It is true that, after performing feature selection with the ARIMA, we can increase the

accuracy of our forecasts by performing ANN, SVR, and MARS modeling with the additional sample data.

6. Conclusion

Humans survive based on food crops. We eat significant quantities of rice, wheat, corn, and other simple crops to maintain energy and good health. Accordingly, food crop price forecasting is very important and has drawn considerable attention in recent decades.

TABLE 19: Improvement of the proposed integrated models over single models.

	MAE _{PI} (%)	RMSE _{PI} (%)	MAPE _{PI} (%)
(1) Rice price forecasting			
(ii) Integrated ARIMA-ANN			
ARIMA	369.822	340.945	376.038
ANN ₁	34.250	27.273	34.226
(ii) Integrated ARIMA-SVR			
ARIMA	511.925	442.372	513.272
SVR ₁	251.772	252.688	236.023
(iii) Integrated ARIMA-MARS			
ARIMA	333.093	240.977	332.613
MARS ₁	81.359	61.186	78.189
(2) Wheat price forecasting			
(i) Integrated ARIMA-ANN			
ARIMA	273.108	288.223	308.313
ANN ₁	30.810	21.562	29.598
(ii) Integrated ARIMA-SVR			
ARIMA	242.775	271.562	283.046
SVR ₁	4.336	19.483	7.223
(iii) Integrated ARIMA-MARS			
ARIMA	214.949	203.152	246.247
MARS ₁	20.304	-1.029	17.842
(3) Corn price forecasting			
(i) Integrated ARIMA-ANN			
ARIMA	102.896	86.124	106.391
ANN ₁	2.018	-10.973	1.645
(ii) Integrated ARIMA-SVR			
ARIMA	105.605	109.089	108.399
SVR ₁	-2.601	-5.525	-3.677
(iii) Integrated ARIMA-MARS			
ARIMA	127.606	120.253	132.074
MARS ₁	8.006	2.579	8.587

TABLE 20: PIs of the whole food crop prices by using the proposed integrated models over the single ARIMA models.

	MAE (%)	RMSE (%)	MAPE (%)
(1) ARIMA-ANN over ARIMA			
Rice	369.822	340.945	376.038
Wheat	273.108	288.223	308.313
Corn	102.896	86.124	106.391
Average PIs	248.609	238.431	263.581
(2) ARIMA-SVR over ARIMA			
Rice	511.925	442.372	513.272
Wheat	242.775	271.562	283.046
Corn	105.605	109.089	108.399
Average PIs	286.768	274.341	301.572
(3) ARIMA-MARS over ARIMA			
Rice	333.093	240.977	332.613
Wheat	214.949	203.152	246.247
Corn	127.606	120.253	132.074
Average PIs	225.216	188.127	236.978

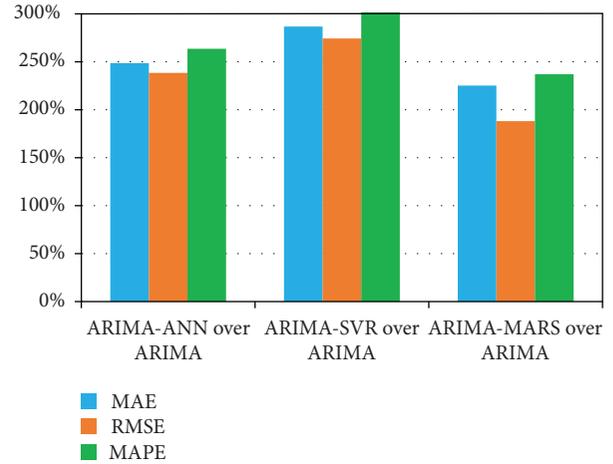


FIGURE 4: Average PIs by using the proposed integrated models over single ARIMA models.

Typical CI forecasting techniques require proper explanatory variables to make predictions. However, proper explanatory variables are difficult to capture, and it is infeasible to

TABLE 21: PIs of the whole food crop prices by using the proposed integrated models over another element, other than ARIMA element.

	MAE (%)	RMSE (%)	MAPE (%)
(1) ARIMA-ANN over ANN			
Rice	34.250	27.273	34.226
Wheat	30.810	21.562	29.598
Corn	2.018	-10.973	1.645
Average PIs	22.359	12.621	21.823
(2) ARIMA-SVR over SVR			
Rice	251.772	252.688	236.023
Wheat	4.336	19.483	7.223
Corn	-2.601	-5.525	-3.677
Average PIs	84.502	88.882	79.856
(3) ARIMA-MARS over MARS			
Rice	81.359	61.186	78.189
Wheat	20.304	-1.029	17.842
Corn	8.006	2.579	8.587
Average PIs	36.556	20.912	34.873

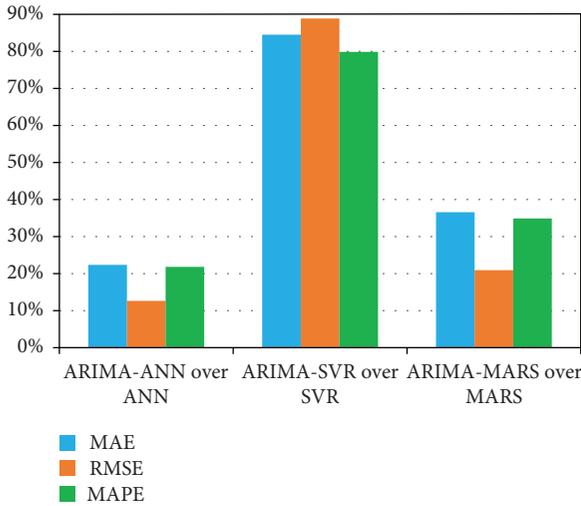


FIGURE 5: Average PIs by using the proposed integrated models over single ANN, SVR, and MARS models.

obtain the future values of these variables. Therefore, we proposed integrated ARIMA-ANN, ARIMA-SVR, and ARIMA-MARS models in order to perform forecasting for three important food crop prices. The role of the ARIMA element is as an FSM that can capture important self-predictor variables. Rather than using unavailable explanatory variables, the self-predictor variables serve as the inputs for ANN, SVR, and MARS modeling. The experimental results reveal that the proposed integrated models are desirable alternatives for food crop price forecasting, because they all have excellent forecasting performance. Most importantly, the main contribution of the proposed models is their ability to provide

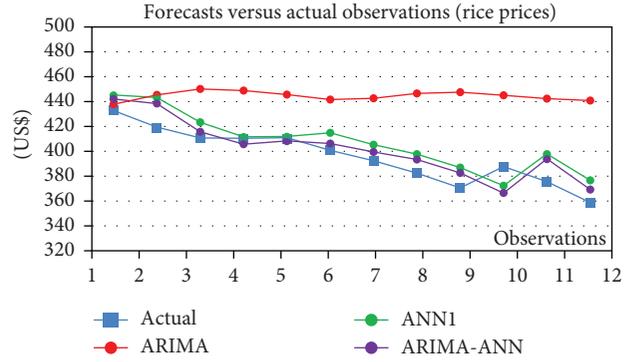


FIGURE 6: Forecasts of rice prices versus actual observations for the 12 testing records (with the use of ARIMA, ANN₁, and the proposed ARIMA-ANN models).

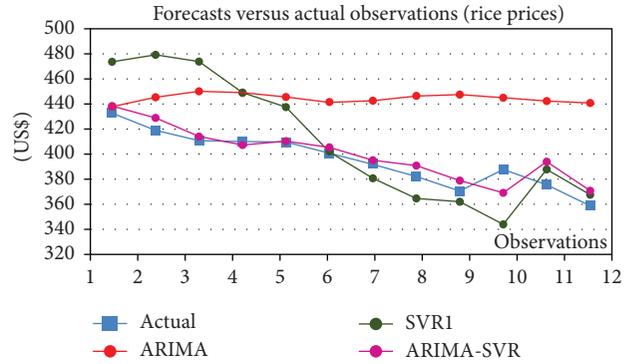


FIGURE 7: Forecasts of rice prices versus actual observations for the 12 testing records (with the use of ARIMA, SVR₁, and the proposed ARIMA-SVR models).

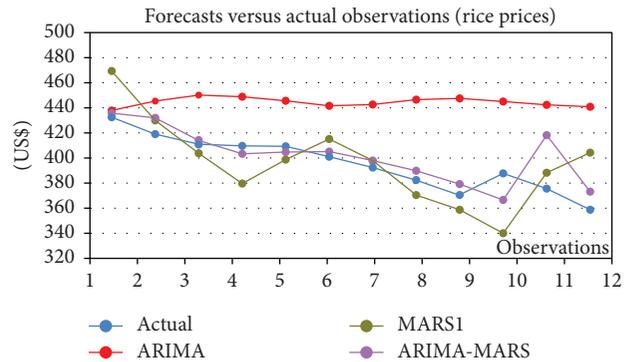


FIGURE 8: Forecasts of rice prices versus actual observations for the 12 testing records (with the use of ARIMA, MARS₁, and the proposed ARIMA-MARS models).

predictions of food crop prices without requiring extensive effort to obtain the future values of explanatory variables.

One limitation of our models is that the techniques for identifying the correct ARIMA model from the variety of possible models may be unintuitive and computationally expensive. However, the modeling in this study can be used as a guideline for developing forecasting models for other

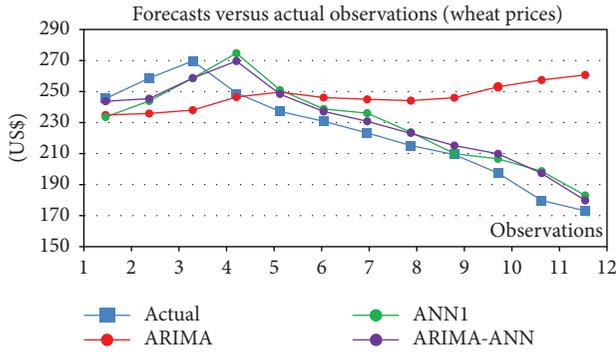


FIGURE 9: Forecasts of wheat prices versus actual observations for the 12 testing records (with the use of ARIMA, ANN₁, and the proposed ARIMA-ANN models).

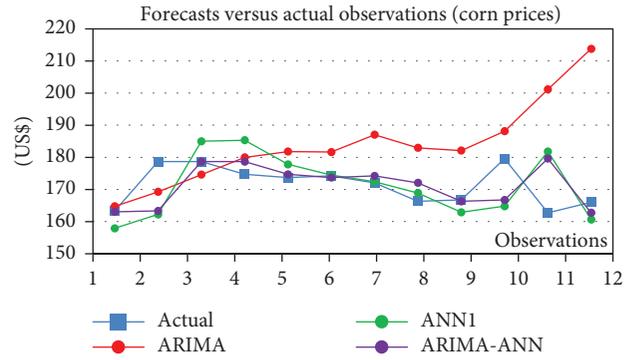


FIGURE 12: Forecasts of corn prices versus actual observations for the 12 testing records (with the use of ARIMA, ANN₁, and the proposed ARIMA-ANN models).

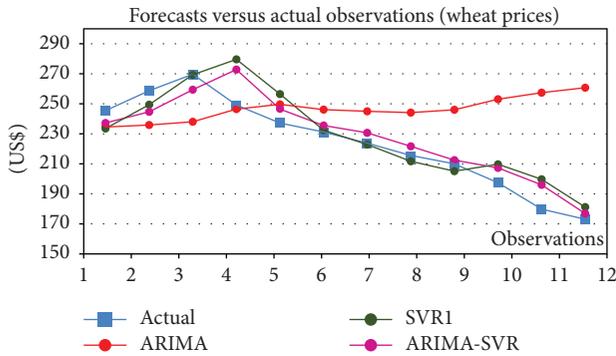


FIGURE 10: Forecasts of wheat prices versus actual observations for the 12 testing records (with the use of ARIMA, SVR₁, and the proposed ARIMA-SVR models).

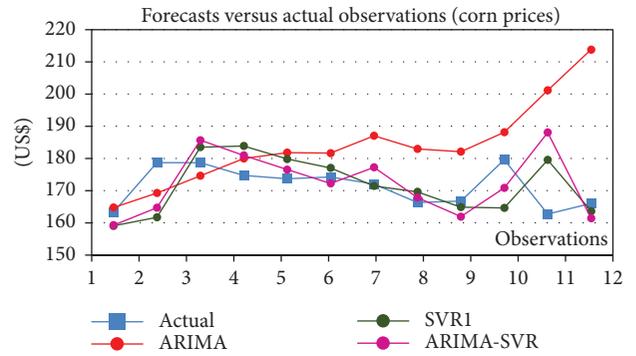


FIGURE 13: Forecasts of corn prices versus actual observations for the 12 testing records (with the use of ARIMA, SVR₁, and the proposed ARIMA-SVR models).

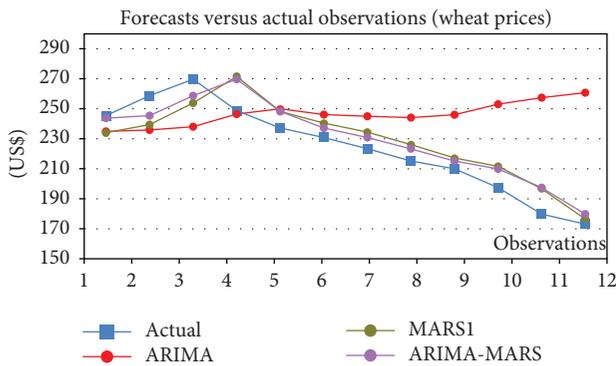


FIGURE 11: Forecasts of wheat prices versus actual observations for the 12 testing records (with the use of ARIMA, MARS₁, and the proposed ARIMA-MARS models).

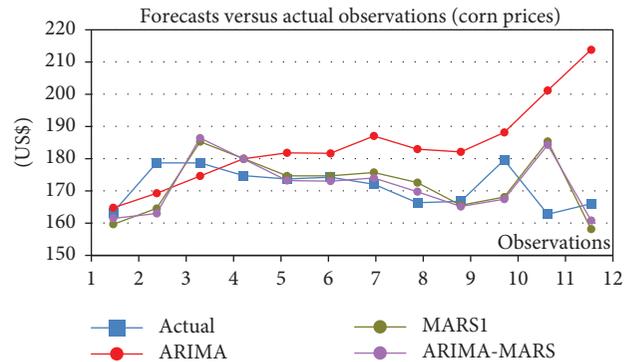


FIGURE 14: Forecasts of corn prices versus actual observations for the 12 testing records (with the use of ARIMA, MARS₁, and the proposed ARIMA-MARS models).

food crop price data series. Additionally, because MARS is effective for selecting important variables for predicting response variables, attempting to extend the FSM of MARS modeling may be a valuable future research direction. Finally, the integrated models may also be combined with other CI techniques, such as extreme learning machines, time

delay neural networks, or artificial immune systems, which may be worthy of investigation in the future.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work is partially supported by the Ministry of Science and Technology of the Republic of China, Grant no. MOST 106-2221-E-030-010-MY2.

References

- [1] H. C. J. Godfray, J. R. Beddington, I. R. Crute et al., "Food security: the challenge of feeding 9 billion people," *Science*, vol. 327, no. 5967, pp. 812–818, 2010.
- [2] D. Tilman, C. Balzer, J. Hill, and B. L. Befort, "Global food demand and the sustainable intensification of agriculture," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 50, pp. 20260–20264, 2011.
- [3] Food and Agriculture Organization of the United Nations, "How to feed the world in 2050," 2009, <http://www.fao.org/wsfs/forum2050/wsfs-forum/en/>.
- [4] International Development Research Centre, "Facts & figures on food and biodiversity," 2017, <https://www.idrc.ca/en/article/facts-figures-food-and-biodiversity>.
- [5] H. C. Co and R. Boosarawongse, "Forecasting Thailand's rice export: statistical techniques vs. artificial neural networks," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 610–627, 2007.
- [6] H. F. Zou, G. P. Xia, F. T. Yang, and H. Y. Wang, "An investigation and comparison of artificial neural network and time series models for Chinese food grain price forecasting," *Neurocomputing*, vol. 70, no. 16–18, pp. 2913–2923, 2007.
- [7] P. A. Seck, E. Tollens, M. C. S. Wopereis, A. Diagne, and I. Bamba, "Rising trends and variability of rice prices: threats and opportunities for sub-Saharan Africa," *Food Policy*, vol. 35, no. 5, pp. 403–411, 2010.
- [8] A. Jumah and R. M. Kunst, "Seasonal prediction of European cereal prices: good forecasts using bad models?," *Journal of Forecasting*, vol. 27, no. 5, pp. 391–406, 2008.
- [9] U. Chakravorty, M. H. Hubert, M. Moreaux, and L. Nøstbakken, "Long-run impact of biofuels on food prices," *The Scandinavian Journal of Economics*, vol. 119, no. 3, pp. 733–767, 2017.
- [10] R. Cai, J. D. Mullen, M. E. Wetzstein, and J. C. Bergstrom, "The impacts of crop yield and price volatility on producers' cropping patterns: a dynamic optimal crop rotation model," *Agricultural Systems*, vol. 116, pp. 52–59, 2013.
- [11] M. Yousefi, B. Khoshnevisan, S. Shamshirband et al., "Support vector regression methodology for prediction of output energy in rice production," *Stochastic Environmental Research and Risk Assessment*, vol. 29, no. 8, pp. 2115–2126, 2015.
- [12] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Systems with Applications*, vol. 37, no. 1, pp. 479–489, 2010.
- [13] Y. E. Shao and C. D. Hou, "Change point determination for a multivariate process using a two-stage hybrid scheme," *Applied Soft Computing*, vol. 13, no. 3, pp. 1520–1527, 2013.
- [14] Y. E. Shao, C. D. Hou, and C. C. Chiu, "Hybrid intelligent modeling schemes for heart disease classification," *Applied Soft Computing*, vol. 14, pp. 47–52, 2014.
- [15] G. Santamaria-Bonfil, A. Reyes-Ballesteros, and C. Gershenson, "Wind speed forecasting for wind farms: a method based on support vector regression," *Renewable Energy*, vol. 85, pp. 790–809, 2016.
- [16] R. Tahmasebifar, M. K. Sheikh-El-Eslami, and R. Kheirollahi, "Point and interval forecasting of real-time and day-ahead electricity prices by a novel hybrid approach," *IET Generation, Transmission & Distribution*, vol. 11, no. 9, pp. 2173–2183, 2017.
- [17] O. Abedinia, N. Amjady, and H. Zareipour, "A new feature selection technique for load and price forecast of electrical power systems," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 62–74, 2017.
- [18] A. R. Gollou and N. Ghadimi, "A new feature selection and hybrid forecast engine for day-ahead price forecasting of electricity markets," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 6, pp. 4031–4045, 2017.
- [19] S. Jurado, A. Nebot, F. Mugica, and N. Avellana, "Hybrid methodologies for electricity load forecasting: entropy-based feature selection with machine learning and soft computing techniques," *Energy*, vol. 86, pp. 276–291, 2015.
- [20] F. Jimenez, G. Sanchez, J. M. Garcia, G. Sciavicco, and L. Miralles, "Multi-objective evolutionary feature selection for online sales forecasting," *Neurocomputing*, vol. 234, pp. 75–92, 2017.
- [21] O. B. Shukur and M. H. Lee, "Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA," *Renewable Energy*, vol. 76, pp. 637–647, 2015.
- [22] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, San Francisco, CA, USA, 1970.
- [23] G. M. Ljung and G. E. P. Box, "On a measure of lack of fit in time series models," *Biometrika*, vol. 65, no. 2, pp. 297–303, 1978.
- [24] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.
- [25] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 2000.
- [26] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks*, vol. 17, no. 1, pp. 113–126, 2004.
- [27] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991.
- [28] 2017, <https://www.indexmundi.com/>.
- [29] Rice, <https://www.indexmundi.com/commodities/?commodity=rice&months=360>.
- [30] Wheat, <https://www.indexmundi.com/commodities/?commodity=wheat&months=360>.
- [31] Corn, <https://www.indexmundi.com/commodities/?commodity=corn&months=360>.
- [32] Y. E. Shao and C. C. Chiu, "Applying emerging soft computing approaches to control chart pattern recognition for an SPC–EPC process," *Neurocomputing*, vol. 201, pp. 19–28, 2016.

Research Article

About Extracting Dynamic Information of Unknown Complex Systems by Neural Networks

Eloy Irigoyen ¹, Antonio Javier Barragán,² Mikel Larrea ¹ and José Manuel Andújar ²

¹UPV/EHU, Alda, Urquijo, s/n, Bizkaia, Spain

²UHU, ETSI, Campus de El Carmen, Huelva, Spain

Correspondence should be addressed to Eloy Irigoyen; eloy.irigoyen@ehu.eus

Received 18 October 2017; Revised 25 April 2018; Accepted 16 May 2018; Published 8 July 2018

Academic Editor: Marek Reformat

Copyright © 2018 Eloy Irigoyen 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 work presents a straightforward methodology based on neural networks (NN) which allows to obtain relevant dynamic information of unknown nonlinear systems. It provides an approach for cases in which the complex task of analyzing the dynamic behaviour of nonlinear systems makes it excessively challenging to obtain an accurate mathematical model. After reviewing the suitability of multilayer perceptrons (MLPs) as universal approximators to replace a mathematical model, the first part of this work presents a system representation using a model formulated with state variables which can be exported to a NN structure. Considering the linearization of the NN model in a mesh of operating points, the second part of this work presents the study of equilibrium states in such points by calculating the Jacobian matrix of the system through the NN model. The results analyzed in three case studies provide representative examples of the strengths of the proposed method. Conclusively, it is feasible to study the system behaviour based on MLPs, which enables the analysis of the local stability of the equilibrium points, as well as the system dynamics in its environment, therefore obtaining valuable information of the system dynamic behaviour.

1. Introduction

The innumerable strategies and new proposals in the control system area are generally based on the knowledge of the system to be controlled. In some cases, its model is achieved by mathematical and analytical procedures, because such model is a simplified representation of the system, or the problem to study is highly restricted to a specific operating point. Although the modelling is often approached by the means of mathematical and analytical methods, with the growing complexity involved in the systems and the requirements of accurate representation, such approaches turn into an increasingly challenging solution.

Consequently, considering the maturity of computational intelligence (CI) techniques and new enabler technologies, CI methods represent an attractive alternative to develop accurate modelling and control solutions for highly complex models [1–4].

Taking into account the need to obtain precise models of complex systems which are applicable in a wide operating

range, the CI techniques are an ideal method to reproduce the behaviour of a complex system [2, 5, 6] or, as it will be presented in this work, to analyze the dynamics of systems.

When studying a process in engineering, in biomedical field, in natural sciences, and even in social systems, approaching the analysis from a dynamic point of view can be very attractive and convenient, depending on the focus of such study. A dynamic analysis of the system can provide wide and very rich information related to how the system will respond under certain inputs. Moreover, it can allow to study its dynamic behaviour through the analysis of the stability in open-loop, both locally and globally. In addition, it will be possible to study whether certain nonlinear phenomena affect the system, for example bifurcations, saddle, and limit cycles [7, 8].

The methods traditionally applied in control engineering are based on linear approximations around several operating points of the system. This is suitable when problems are studied and solved in a local domain. However, there is a

trend to approach bigger problems with a more abstracted and global perspective, leading to the use of nonlinear methods [9–12]. At some point of complexity, and certainly if the system involves unknown parts and other uncertainties, an entirely formal analysis of the system becomes unsustainable [13–15].

One of the main reasons for the use of nonlinear models is based on the dynamics of linear systems, since conventional mathematical formulations are not rich enough to reproduce a series of phenomena that usually appear in the real life [2, 16, 17]. The dynamic behaviour of a linear system, without considering its order, is basically governed by the eigenvalues of the corresponding state matrix [8]. On the contrary, nonlinear systems show a much richer behaviour, with self-excited oscillations (i.e., limit cycles), aperiodic behaviours and critically sensitive to the initial conditions [7], and chaos [18], as well as other dynamic phenomena exclusive to nonlinear systems, such as the existence of multiple states of equilibrium and bifurcations [19], among others.

The typically appropriate initial approach to analyze nonlinear systems is to use a representation of the system by means of a mathematical model, generally represented in state variables. This is possible assuming that sufficient information and knowledge of the system is available to generate its state equations, provided that the system dynamic is not extremely complex. In many applications, current research deals with the study of unknown complex systems, whether due to a complex dynamic, high dimensionality, or lack of information about the physical relationships that govern the behaviour of the system. In such situations, the techniques from the field of intelligent control can help to improve these studies, as Barragán et al. present in [20, 21] using Fuzzy logic to define a formal methodology for analyzing the dynamic behaviour of nonlinear systems, or Grande et al. in [11] to extract qualitative models of spatial evolution from a chemical system. In the same sense, neural networks (NN) become a powerful technique, since they are able to model highly complex nonlinear systems from input-output data. Proper selection and training of a basic structure such as a multilayer perceptron (MLP) can accurately reproduce the behaviour of a nonlinear system. This modelling technique can be used, both qualitatively and analytically [6, 22–25], taking into account that MLPs are universal approximators, either for a function [26–29] or its derivative [30, 31]. Thus, although the system might be unknown, it is possible to obtain a NN model of its behaviour, representing its dynamics in the workspace studied. In a formal sense, a NN model is a mathematical model. Hence, from this NN model, it is possible to study several aspects of the real dynamics of the system, conditioned only by the high precision of the model. This can be achieved with an exhaustive experimental stage where the topology of a NN is selected, capable of faithfully reproducing the behaviour of the system with the expected precision. This approach in solving this problem allows dealing with nonlinear systems, where modeling by traditional mathematical techniques can be challenging.

During the analysis and design of control solutions, knowing the equilibrium states of a system, as well as the

stability of such states, is an aspect of great interest. When the model of the system is completely unknown, this information could help to clarify how the system works, even to ease the design of an appropriate control. It should be noted that despite the existence of recent works that present formal analysis methodologies based on Fuzzy logic [17, 20, 21, 32] the authors have not currently found any work focused from the NN point of view, under a general approach as presented in this proposal.

This work presents a straightforward and easy to use methodology for extracting information from unknown systems using NNs. The main objective of this proposal is to develop a method that allows obtaining information on the dynamics of nonlinear systems, when there is no mathematical model, neither accurate nor approximate, to analyze them. In these situations, any additional information reached by new methods is significant, especially when this information is related to the analysis of the presence of equilibrium states and their local stability, as presented below. In this work, an MLP neural network is trained with a set of measured values of inputs and outputs of supposedly unknown systems, in order to reproduce the behaviour of these systems. For this purpose, taking into account that the dynamic study aims at analyzing the behaviour of each nonlinear system in their corresponding equilibrium states, the dataset of examples to train the NN will be obtained from its entire operating range. More specifically, the equilibrium states of three nonlinear systems will be studied through their NN models, which reproduce their corresponding state variable models. The equilibrium states are reached by a precise linearization in a grid of operating points extracted from the NN models, and subsequently performing a study of local stability. Using this information, the local stability of equilibrium states is obtained, as well as the system dynamics in the vicinity of the studied points, achieving valuable information about the dynamic behaviour of the nonlinear system.

This paper is organized as follows: Section 2 presents the problem and the formulation associated with it, explaining how it will be dealt with throughout the document. In Section 3, the procedures to obtain the linearization of a system and its extension to a NN model are explained. Section 4 presents three case studies to demonstrate the proposed approach, based on solid results. Finally, this work finalizes with the corresponding conclusions.

2. Problem Formulation

A generic continuous dynamic system will be considered, represented by state variables $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$, where $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$ and $\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T$ depict a m input system of order n with $u_i(t)$ representing the inputs and $x_i(t)$ the state variables. \mathbf{f} is a static nonlinear map defined as $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ [33]. An equivalent NN model, based on a MLP structure, which can estimate both the state variables of the continuous system $\hat{\mathbf{x}}(t)$ and the system output $\hat{\mathbf{y}}(t)$, is represented by the equations in (1) [34–36].

$$\begin{aligned}\widehat{\mathbf{x}}(t|\boldsymbol{\theta}) &= \mathbf{g}(\boldsymbol{\varphi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}), \\ \widehat{\mathbf{y}}(t|\boldsymbol{\theta}) &= \mathbf{C}(\boldsymbol{\theta})\widehat{\mathbf{x}}(t|\boldsymbol{\theta}),\end{aligned}\quad (1)$$

$$\text{being } \boldsymbol{\varphi}(t, \boldsymbol{\theta}) = \begin{bmatrix} \mathbf{x}(t|\boldsymbol{\theta}) \\ \mathbf{u}(t) \end{bmatrix}, \quad (2)$$

where $\boldsymbol{\varphi}(t, \boldsymbol{\theta})$ is the regression vector and $\boldsymbol{\theta}$ the vector of parameters (\mathbf{w}, \mathbf{W}) of the NN, the inner weights, and the biases. $\mathbf{g}(\cdot)$ is the function realized by the MLP, defined as $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$. To model the evolution of each i th state variable, a MLP structure has to be trained adapting the parameters mentioned above. In the training process, the needed information is provided for both the states $x_i(t)$, with $i = 1, \dots, n$, and the system inputs $u_j(t)$, with $j = 1, \dots, m$.

By selecting a simple MLP structure that consists of one hidden layer of h neurons, with sigmoidal activation functions, and a linear output layer, the NN output O_{MLP} can be calculated by the following general expression [35]:

$$O_{MLP} = \sum_{h=1}^{n_h} W_{k,h} F_h(act_h) + W_{k,0}, \quad (3)$$

where n_h is the number of hidden neurons, k represents the output neuron (for the case of several NN outputs), $W_{k,h}$ are the output layer weights and biases, F_h is the activation function of each hidden neuron, and act_h is the sum of weighted inputs to each hidden neuron, as shown in

$$act_h = \sum_{l=1}^{n+m} \omega_{h,l} \varphi_l(t) + \omega_{h,0}, \quad (4)$$

where $\omega_{h,l}$ are the weights and biases of the hidden layer and $\varphi_l(t)$ is the input vector to the NN, being $n+m$ the vector dimension.

Taking into account that a NN will reproduce the evolution of each state variable $\dot{x}_i(t)$, the neural model can be related to the state model [35] as

$$\dot{x}_i(t) = f_i(\mathbf{x}(t), \mathbf{u}(t)) \equiv \widehat{\dot{x}}_i(t|\boldsymbol{\theta}), \quad (5)$$

$$\widehat{\dot{x}}_i(t|\boldsymbol{\theta}) = \sum_{h=1}^{n_h} W_{i,h} F_h \left[\sum_{l=1}^{n+m} \omega_{h,l} \varphi_l(t) + \omega_{h,0} \right] + W_{i,0}. \quad (6)$$

From the above representation, in order to simplify the methodology in studying the obtained NN models, each state variable will be modeled by a different NN.

3. Information Obtained from the Neural Network Model

After obtaining an accurate model of a system, it is a fact that this model can be used to obtain system information through well-known techniques. In this section, a very important technique is presented to study nonlinear systems in two phases, as required by the methodology of this work. Firstly, the linearization of a neural state model will be exposed in detail. Secondly, the study of the equilibrium states of an unknown nonlinear system will be presented. This study is

carried out through a NN model that reproduces the behaviour of the aforementioned nonlinear system. The study of the equilibrium states from the NN model, together with the study of their local stability from the linearization, allows to analyze the operational behaviour of a system from a qualitative point of view.

3.1. Linearization of a Neural Model. Linearization is one of the most commonly used techniques in solving design problems in the field of nonlinear control systems, even though it is necessary to point out that this is a technique not ideal in many situations where the effects of nonlinearities are not negligible. It is a very convenient technique for the control of not excessively complex systems or in situations when the dynamics of the system is approximately known in regions where the system behaviour is close to a linear one, basically around equilibrium states.

Thereby, apart from being a method that aims at the control of systems, linearization could be a powerful resource to obtain information from a nonlinear system. It could be considered that, except in some situations, the behaviour of a nonlinear system around an equilibrium state is analogous to the one observed after linearization of the system in such state [19, 37, 38]. So, the study and calculation of equivalent linear systems from a nonlinear NN model can be a powerful technique to obtain information concerning the real nonlinear system analyzed.

The generic state model, obtained from a nonlinear system, is represented by

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ \text{where } \mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m &\longrightarrow \mathbb{R}^n.\end{aligned}\quad (7)$$

The first-order simplification of the Taylor series of the nonlinear system, in the domain of the state $(\mathbf{x}_0, \mathbf{u}_0)$, can be determined an approximation as

$$\dot{\mathbf{x}}(t) \approx \mathbf{A}_0 + \mathbf{A}\bar{\mathbf{x}}(t) + \mathbf{B}\bar{\mathbf{u}}(t), \quad (8)$$

where \mathbf{A}_0 , \mathbf{A} , and \mathbf{B} are $(n \times 1)$, $(n \times n)$, and $(n \times m)$ matrices, respectively.

Being $\bar{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_0$ and $\bar{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_0$, the matrices of the linearized system are obtained as

$$\begin{aligned}\mathbf{A}_0 &= \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0), \\ \mathbf{A} &= \left(\begin{array}{ccc} \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_1} & \dots & \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial x_n} \end{array} \right) \Bigg|_{(\mathbf{x}_0, \mathbf{u}_0)}, \\ \mathbf{B} &= \left(\begin{array}{ccc} \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_1} & \dots & \frac{\partial f_1(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_1} & \dots & \frac{\partial f_n(\mathbf{x}(t), \mathbf{u}(t))}{\partial u_m} \end{array} \right) \Bigg|_{(\mathbf{x}_0, \mathbf{u}_0)}.\end{aligned}\quad (9)$$

For the rest of the presented work, the time (t) dependence of state variables and system inputs will be suppressed in order to abbreviate the expressions.

If $(\mathbf{x}_0, \mathbf{u}_0)$ is an equilibrium state of the system, the matrix \mathbf{A}_0 will be zero, since by definition, an equilibrium state leads the state equation to zero.

When the system (7) is represented by a NN model, the equivalent mathematical model is shown as (6). Linearizing the (7) around the state $(\mathbf{x}_0, \mathbf{u}_0)$, the new equivalent mathematical model of a linearized neural model is represented by (10), being x_q and u_v the q th and v th vector components of \mathbf{x}_0 and \mathbf{u}_0 , respectively. These components are also the set of inputs ($n+m$) of the NN.

$$\dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left. \frac{\partial f_i}{\partial x_q} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q + \sum_{v=n+1}^{n+m} \left. \frac{\partial f_i}{\partial u_v} \right|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v. \quad (10)$$

By extending the previous (10) with (6),

$$\begin{aligned} \dot{x}_i \approx & f_i(\mathbf{x}_0, \mathbf{u}_0) \\ & + \sum_{q=1}^n \left[\frac{\partial}{\partial x_q} \left(\sum_{h=1}^{n_h} W_{i,h} F_h \left[\sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right. \right. \\ & \left. \left. + W_{i,0} \right) \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q \\ & + \sum_{v=n+1}^{n+m} \left[\frac{\partial}{\partial u_v} \left(\sum_{h=1}^{n_h} W_{i,h} F_h \left[\sum_{l=n+1}^{n+m} \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right. \right. \\ & \left. \left. + W_{i,0} \right) \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v, \end{aligned} \quad (11)$$

and consequently, (12) is obtained as

$$\begin{aligned} \dot{x}_i \approx & f_i(\mathbf{x}_0, \mathbf{u}_0) \\ & + \sum_{q=1}^n \left[\frac{\partial \left(\sum_{h=1}^{n_h} W_{i,h} F_h \left[\sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right)}{\partial x_q} \right] \Big|_{\bar{x}_q(\mathbf{x}_0, \mathbf{u}_0)} \\ & + \sum_{v=n+1}^{n+m} \left[\frac{\partial \left(\sum_{h=1}^{n_h} W_{i,h} F_h \left[\sum_{l=n+1}^{n+m} \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right)}{\partial u_v} \right] \Big|_{\bar{u}_v(\mathbf{x}_0, \mathbf{u}_0)}, \end{aligned} \quad (12)$$

in a reduced form to work with

$$\dot{x}_i \approx f_i(\mathbf{x}_0, \mathbf{u}_0) + \sum_{q=1}^n \left[\frac{\partial G_h}{\partial x_q} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q + \sum_{v=n+1}^{n+m} \left[\frac{\partial H_h}{\partial u_v} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v. \quad (13)$$

Subsequently, based on the works of Pirabakaran and Becerra [23] and Larrea [39], where these derivatives are calculated through the internal connections of the NN, (13) can be calculated separately for each of its terms. For this purpose, it is necessary to define the activation function of the hidden layer neurons, $F_j(activ_h)$. In this work, for the neurons of the hidden layer of the selected multilayer perceptron structure, the activation function hyperbolic tangent is chosen. The selection of a smooth activation function is performed to enable the calculation of the partial derivatives

shown in (16). Applying the chain rule, we decompose the first part in three partial derivatives.

$$\frac{\partial G_h}{\partial x_q} = \frac{\partial G_h}{\partial F_h} \frac{\partial F_h}{\partial activ_h} \frac{\partial activ_h}{\partial x_q}. \quad (14)$$

The first partial derivative of (14) is straightforward.

$$\frac{\partial G_h}{\partial F_h} = W_{i,h}. \quad (15)$$

As the activation function $F_h(\cdot)$ is $\tanh(\cdot)$, we obtain that the second partial derivative of (14) results in

$$\frac{\partial F_h(activ_h)}{\partial activ_h} = \frac{\partial}{\partial activ_h} \frac{\sinh(activ_h)}{\cosh(activ_h)} = 1 - \tanh^2(activ_h) = 1 - o_h^2, \quad (16)$$

where o_h is precisely the value of the h th neuron output for the MLP neural network, $\tanh(activ_h) = o_h$, whilst the third partial derivative in (14) results in

$$\frac{\partial^2 activ_h}{\partial x_q^2} = \frac{\partial}{\partial x_q} \sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} = \omega_{h,q}. \quad (17)$$

Then, the overall solution for $\partial G_h / \partial x_q$ is

$$\frac{\partial \left(\sum_{h=1}^{n_h} W_{i,h} F_h \left[\sum_{l=1}^n \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right)}{\partial x_q} = \sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,q}, \quad (18)$$

being q an integer value into the interval $[1, n]$.

For the second derivative $\partial H_h / \partial u_v$, the procedure is similar to the previous one, obtaining the expression

$$\frac{\partial}{\partial u_v} \left(\sum_{h=1}^{n_h} W_{i,h} F_h \left[\sum_{l=n+1}^{n+m} \omega_{h,l\varphi l}(t) + \omega_{h,0} \right] \right) = \sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,v}, \quad (19)$$

where v has integer values into the interval $[n+1, n+m]$.

Substituting (18) and (19) in (12), the equivalent mathematical model of a nonlinear system based on NN and linearization around a state $(\mathbf{x}_0, \mathbf{u}_0)$ is depicted by

$$\begin{aligned} \dot{x}_i \approx & f_i(\mathbf{x}_0, \mathbf{u}_0) \\ & + \sum_{q=1}^n \left[\sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,q} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{x}_q \\ & + \sum_{v=n+1}^{n+m} \left[\sum_{h=1}^{n_h} W_{i,h} (1 - o_h^2) \omega_{h,v} \right] \Big|_{(\mathbf{x}_0, \mathbf{u}_0)} \bar{u}_v. \end{aligned} \quad (20)$$

3.2. Equilibrium States and Local Stability. In order to perform an exhaustive analysis of the nonlinear system, it is first necessary to obtain an appropriate neural state model of that system, as presented in (1). This could be done through some of the existing modelling techniques [35], either online [40, 41] or offline [1, 6]. Subsequently, it is important to locate the different equilibrium states of the system. The search and location of such equilibrium states

in a control system are one of the first problems that have to be solved in order to develop a study of the behaviour of the system. Before designing the control system, the identification and analysis of the equilibrium states provide valuable information about the behaviour of the system, especially in the case of a nonlinear system, since these states are the most relevant cases to study such systems through linearization techniques. In order to locate the equilibrium states of the system, (21) must be solved.

$$\left[\hat{\mathbf{x}}_1(t) \cdots \hat{\mathbf{x}}_n(t) \right]^T = [0 \cdots 0]^T. \quad (21)$$

For nonlinear dynamics, the equilibrium states could be very difficult to solve analytically, so it is necessary to use numerical methods [26–28, 30]. Given the mathematical model of a NN system characterized by a MLP, see (3), the set of nonlinear equations to solve is the following:

$$\begin{aligned} \sum_{h=1}^{n_h} W_{1,h} F_h(\text{act}_h) + W_{1,0} &= 0 \\ \sum_{h=1}^{n_h} W_{2,h} F_h(\text{act}_h) + W_{2,0} &= 0 \\ &\vdots \\ \sum_{h=1}^{n_h} W_{n,h} F_h(\text{act}_h) + W_{n,0} &= 0. \end{aligned} \quad (22)$$

These equations represent the n MLPs that model the nonlinear system under study. When analyzing the dynamics of nonlinear systems, the calculation of their equilibrium states can be a notable problem. In contrast to linear systems, where one or infinite equilibrium states exist, a nonlinear system can contain one, none, a finite number, or infinite states of equilibrium. For the resolution of the set of equations in (22), both numerical or more complex methods can be utilized. Complex methods, as bioinspired algorithms (i.e., evolutionary computation techniques), can locate a large number of solutions, but its slower convergence is a clear disadvantage in comparison with numerical methods. Then, in order to solve the set of nonlinear equations in (22), the use of numerical methods will be proposed, since they can offer a rapid convergence and precision in the obtained results [42, 43]. In this sense, the Levenberg-Marquardt (L-M) method [44] with the extension proposed by Moré [45] will be performed. This algorithm needs initial conditions to initialize the search. Thus, taking into account that it will be necessary to maximize the probability of finding every existing equilibrium state, a thin grid of points in the ranges of inputs and state variables will be used. Although this is the methodology that has been used in this article, the important thing in this case is to obtain as many solutions as possible, regardless of the method used to find them. Therefore, any other algorithm to solve (22) would be perfectly valid.

The L-M algorithm requires the Jacobian matrix of the system, in order to accelerate its convergence. This matrix can be obtained, either with explicit calculation or with some

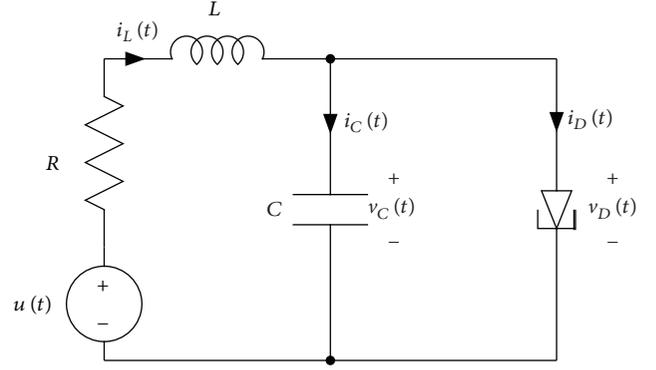


FIGURE 1: Tunnel-diode circuit.

technique to approximate it. In the previous Section 3.1, the calculation of the Jacobian matrix of a NN model has been solved under a general approach. Therefore, this matrix can be included into the numerical algorithm to enhance its precision and velocity of convergence.

Furthermore, the Jacobian matrix can be used, both for solving and finding the equilibrium states and for the linearization of the system in each of the solutions obtained. In this way, it is also possible to study the characteristics of the located equilibrium states, from the eigenvalues of the dynamic matrix of the linearized system. This analysis could improve the interpretation of system dynamics; it could help to study the local stability, even to observe more complex behaviours, such as bifurcations, saddle points, or limit cycles.

4. Case Studies

In this section, three different examples are presented. These examples come from different areas, being nonlinear electrical, mechanical, and biological systems, which initially will be considered as unknown. The algorithms have been implemented with the tools of MATLAB R, both for the MLP neural network training and for the calculation of the NN model linearization.

4.1. Equilibrium States of a Tunnel-Diode Circuit. Let the tunnel-diode circuit shown in Figure 1, where $R = 1, 5 \text{ k}\Omega$, $C = 2 \text{ pF}$, and $L = 5 \mu\text{H}$, with $x_1(t) = v_C(t)$ and $x_2(t) = i_L(t)$ as variables of the system, and $h(v)$ the nonlinear relation between both. This is a case of study broadly used [20, 21], with the state model expressed by (23).

$$\begin{aligned} \dot{x}_1 &= 0.5[-h(x_1) + x_2], \\ \dot{x}_2 &= 0.2(-x_1 - 1.5x_2 + 1.2). \end{aligned} \quad (23)$$

Supposing that $h(v)$ has a nonlinear relation characterized by (24) and shown in Figure 2,

$$\begin{aligned} h(x_1) &= 17.76x_1 - 103.79x_1^2 + 229.62x_1^3 \\ &\quad - 226.31x_1^4 + 83.72x_1^5. \end{aligned} \quad (24)$$

Considering that the system dynamics is unknown, the first stage in the work plan will be to obtain a NN model of

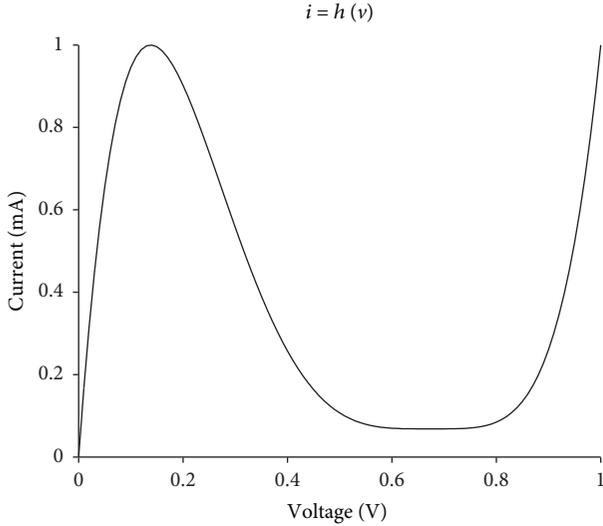


FIGURE 2: Phase portrait of the tunnel-diode circuit.

the system from input-output data. These data will come from an exhaustive selection of examples that will cover the whole universe of the discourse of operating points of the system. Subsequently, after analyzing the NN model and its Jacobian matrix, relevant conclusions will be drawn regarding the dynamics of the system.

In the case of the tunnel-diode, a set of 2000 points is created to model the behaviour of the nonlinear relation $v_D - i_D$ which is uniformly distributed along their discourse universes, respectively.

$$\begin{aligned} X &\subseteq \{x|x_1(t) \in [0, 1], x_2(t) \in [0, 2]\}, \\ U &\subseteq \{u|u(t) \in [0, 2]\}, \end{aligned} \quad (25)$$

where 1600 data samples (80%) will be used in the modelling phase and 400 data samples (20%) in the validation phase. It must be noted that the mathematical model of the tunnel-diode has only been utilized to obtain the dataset, meaning that for the rest of the process it is considered unknown.

In order to keep an appropriate methodological procedure that provides the most adequate MLP neural network, a batch of training experiments has been prepared where the number of neurons in the hidden layer changes, being this the parameter to be studied in relation to the precise response of the NN output. All training experiments have been initialized with an extension of an advanced algorithm to assure the initial local stability of the NN [46]. The final structure selected has been the simplest one obtained that allows to estimate both outputs with enough precision approaching the real ones. This is a MLP with three inputs, x_1 , x_2 , and u , a hidden layer of three neurons with hyperbolic tangent activation function, and two outputs, x_1 and x_2 . The training process was performed with a second order algorithm based on the L-M approach [44, 47], obtaining a validation MSE of 0.1147 V/ns and 3.03 mA/ns, respectively.

To solve the set of equations shown in (22), a minimization algorithm based on the same approach of L-M, adapted by Moré, has been used [44, 45]. With this purpose, a dense mesh has been created to initialize the calculation process whose limits are directly related to the universe of discourse of the studied variables, applying increments of 0.05 and 0.1. In Figure 3, solutions obtained by the numerical algorithm in the NN model are compared with the real system, neglecting the nonvalid solutions located outside the universe of discourse.

Comprehensively using the linearization proposed in this work, it has been possible to find linearized models in each of the equilibrium states shown in Figure 3. These have been characterized in relation to their behaviour, based on the eigenvalues of the dynamic matrix of the linearized system. As can be seen in Figure 3, the equilibrium states of the real system have been calculated accurately, where each type has correlation with the corresponding one obtained by the NN model. It must be noted that the numerical algorithm has found a set of equilibrium states in the NN model which do not exist in the real system. This error may be due to the intrinsic modeling error or to the minimization process with the L-M algorithm. When using numerical methods, tolerances have to be assumed to find the solutions. These tolerances, in addition to the minimization algorithm error, are considered as actual solutions. However, as they are not actual solutions, the study of the eigenvalues in these points reflects an erratic behaviour, since the types of detected eigenvalues are not coherent (at very close points, several types of eigenvalues are detected). This phenomenon does not happen in actual equilibrium states, so it is a simple way to discard the false states detected.

4.2. *The Case of an Inverted Pendulum with Friction.* The second example is an inverted pendulum with friction presented in Figure 4. The dynamics of this system is represented by (26).

$$ML^2\ddot{\theta}(t) + b\dot{\theta}(t) - MgL \sin(\theta(t)) = \tau(t), \quad (26)$$

where the coefficient of friction is $b=0.5$ N/m/s, θ is the pendulum angle with respect to the vertical, τ is the applied pair in N/m, $M=1$ Kg is the mass of the pendulum, the length of rod is $L=0.5$ m, and the acceleration of gravity is $g=9.81$ m/s².

Let be the state vector given by

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} \theta(t) \\ \dot{\theta}(t) \end{pmatrix}. \quad (27)$$

Then, the system of inverted pendulum can be defined by the state model as

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= \frac{g}{L} \sin(x_1(t)) - \frac{b}{ML^2}x_2(t) + \frac{1}{ML^2}u(t). \end{aligned} \quad (28)$$

As a starting point, we consider that the system dynamics is unknown with the aim of validating the techniques

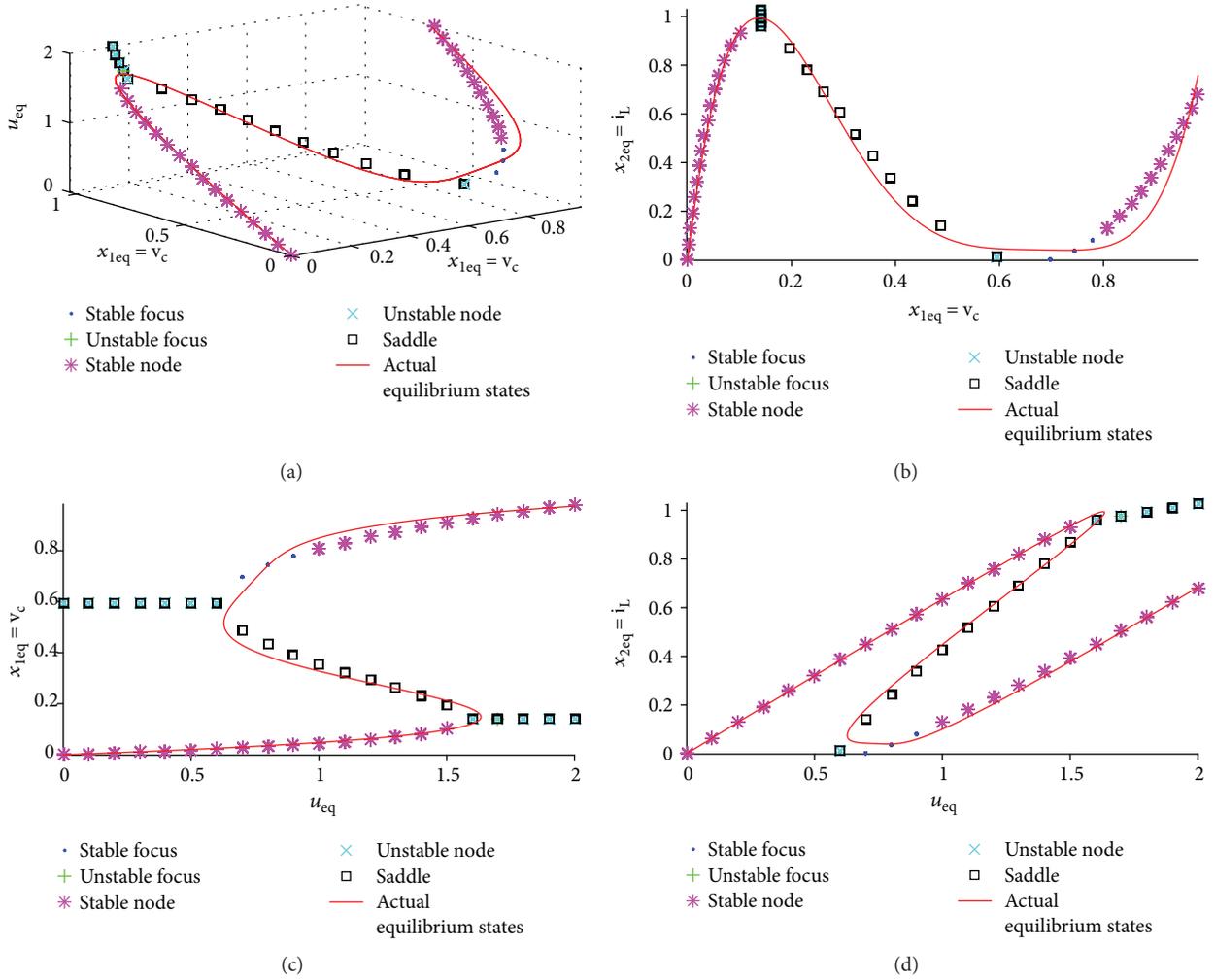


FIGURE 3: Equilibrium states of the tunnel-diode NN model.

developed in this paper. Moreover, we assume that it is possible to collect enough input-output data to obtain the neural model of the system. Thus, the first stage of this work will be to extract a MLP neural model of the pendulum system to be studied.

According to the previous example, a dataset has been prepared considering the universes of discourse of the state variables and input signal, respectively,

$$\begin{aligned} X &\subseteq \{ \mathbf{x} \mid x_1(t) \in [-\pi, \pi], x_2(t) \in [-10, 10] \}, \\ U &\subseteq \{ \tau \mid \tau(t) \in [-10^3, 10^3] \}. \end{aligned} \quad (29)$$

From these discourse universes, 2000 input-output data samples have been extracted, proportionally distributed, taking 1600 data samples (80%) for neural network training and 400 data samples (20%) for the validation. As was the case of the above study, the pendulum system will be considered as unknown.

Also as the previous case, the MLP neural network has been trained using the L-M algorithm [44, 47]. The most appropriate NN structure obtained has three inputs, x_1 , x_2 , and τ , a hidden layer of three neurons with hyperbolic tangent as activation function, and two outputs, \dot{x}_1 and \dot{x}_2 .

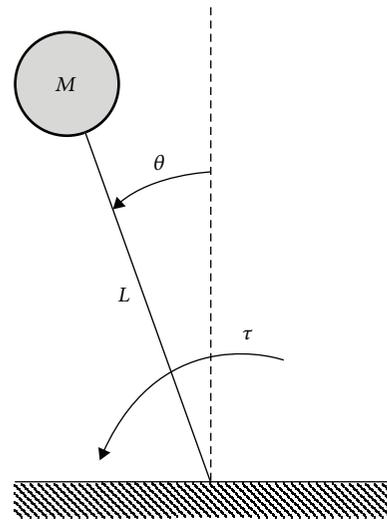


FIGURE 4: Inverted pendulum with friction.

The training process has been performed with the same algorithm, Levenberg-Marquardt, and the validation MSE errors have been 6.334×10^{-6} rad/s and 0.1857 rad.

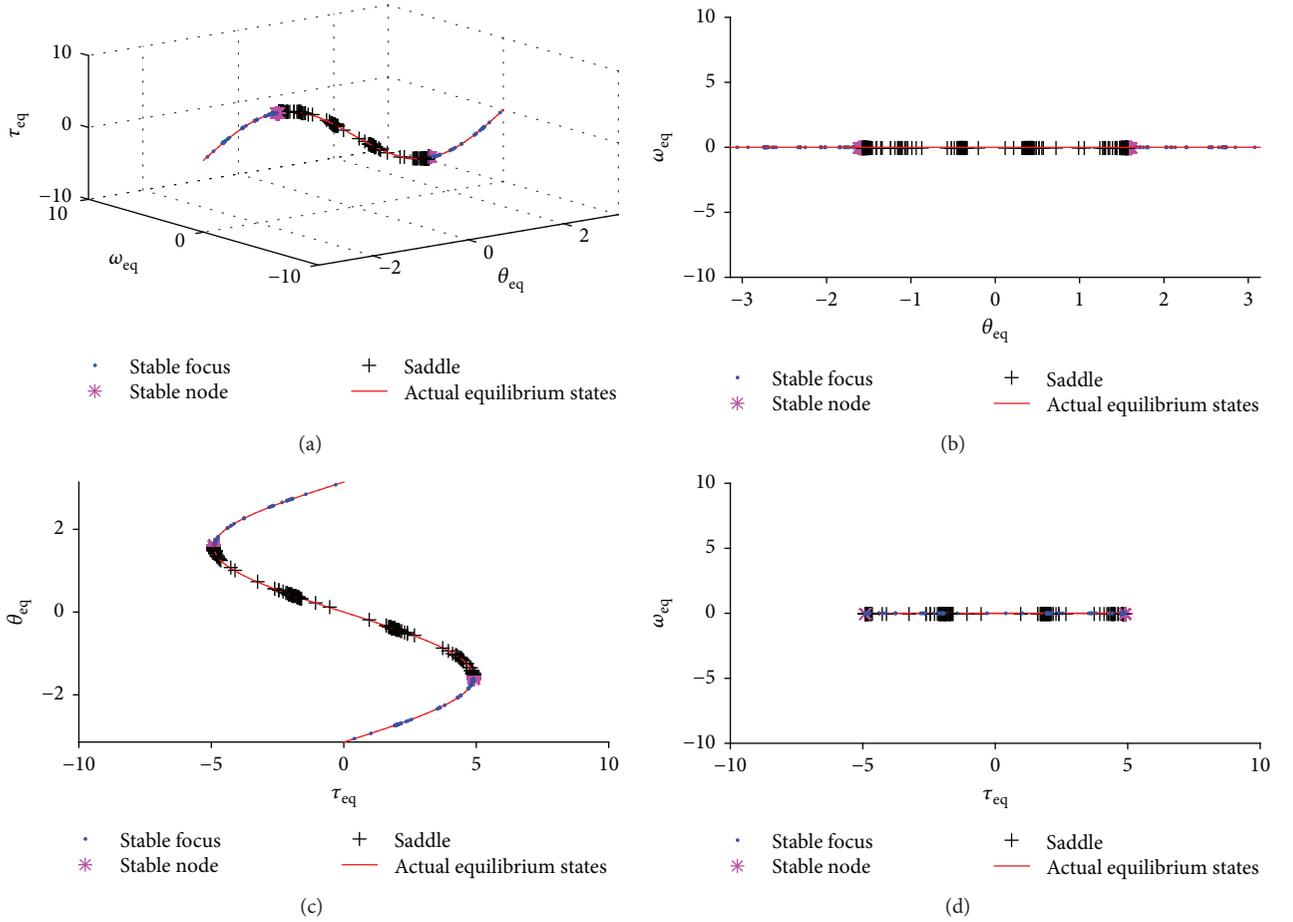


FIGURE 5: Equilibrium states of the inverted pendulum with friction.

Taking into account the obtained NN model of the system, the equilibrium states have been calculated as presented in Section 3.2 and explained at the above example. Thus, a grid of initial points has been created again, limiting each variable range into their universes of discourse, with increments of 0.1, 5, and 4, for the states and the input, respectively. All the equilibrium states located with the NN model are shown in Figure 5. As in the previous example, the points located outside of the universe of discourse have been discarded and removed.

For the real system, its equilibrium states are located on the line $\tau = -MLg \sin(\theta)$, being saddle points the equilibrium points on $-\pi/2 < \theta < \pi/2$, and stable focuses on the rest sited on $[-\pi, \pi]$ [48]. Compared with the equilibrium points estimated by the NN model, it can be concluded that those points have been correctly obtained. Certain discrepancies can be noticed in the steady states obtained around $|\tau| \approx 5$ and $|\theta| \approx \pi/2$, but the rest of equilibrium states estimated into the universe of discourse have been accurately located. The appearance of errors is inevitable as the NN model has an implicit error associated. Considering that such errors are within permissible limits, it is necessary to emphasize that the extracted information, with this neural approach, through the analysis of the

neural structure is a powerful technique to obtain information from an unknown system.

4.3. A Prey-Predator Biological System. Nature is capable of providing diverse real-life examples in which animal behaviour presents multiple stable equilibrium states. This is the case of a predator-prey system with adaptation of prey behaviour to changing environmental conditions [49, 50]. The classic Lotka-Volterra predator-prey model is given by

$$\begin{aligned} \dot{p}(t) &= rp(t)(1 - p(t)/K)ap(t)q(t), \\ \dot{q}(t) &= ep(t)q(t) - mq(t), \end{aligned} \quad (30)$$

where $p(t)$ and $q(t)$ are sizes of the prey and the predator populations, r is per capita reproduction rate of the prey, K is the carrying capacity of environment, m is the per capita mortality rate of the predator in the prey deficiency, a is the attack rate, and $e = ka$ shows the consumption efficiency after the attack. Let be the state vector given by

$$\mathbf{x}(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} p(t) \\ q(t) \end{pmatrix}. \quad (31)$$

Then, the system of a biological prey-predator can be defined by the state model.

$$\begin{aligned}\dot{x}_1(t) &= rx_1(t)(1 - x_1(t)/K) - ax_1(t)x_2(t), \\ \dot{x}_2(t) &= ex_1(t)x_2(t) - mx_2(t).\end{aligned}\quad (32)$$

In response to an external challenge, each individual into the prey community may modify its behaviour between risky and safe conduct. On the basis of the vulnerability of individuals in risky mode, the first expression in (32), suppressing the time (t) dependence, will be extended as

$$\begin{aligned}\dot{x}_{1R} &= r_R x_{1R} - c_R x_{1R}^2 - h_R x_{1R} x_{1S} - a_R x_{1R} x_2, \\ \dot{x}_{1S} &= r_S x_{1S} - c_S x_{1S}^2 - h_S x_{1R} x_{1S} - a_S x_{1S} x_2.\end{aligned}\quad (33)$$

In (33), x_{1R} and x_{1S} are the subpopulations of x_1 in risky (R) and safe (S) modes. In addition, $c_i = r_i/K_i$, h_R , and h_S are magnitudes of the intermode competition [50].

Taking into account that $x_1 = x_{1R} + x_{1S}$, and defining D_S the nonlinear function of the probability for a prey to be hunted by a predator as shown in Figure 6, with $D_R + D_S = 1$ [50], the state model of the behavioural predator-prey system will be

$$\begin{aligned}\dot{x}_1 &= (r_S D_S + r_R(1 - D_S))x_1 - (a_S D_S + a_R(1 - D_S))x_1 x_2 \\ &\quad - (c_S D_S^2 + c_R(1 - D_S)^2 + (h_R + h_S)D_S(1 - D_S))x_1^2, \\ \dot{x}_2 &= (e_S D_S + e_R(1 - D_S))x_1 x_2 - mx_2.\end{aligned}\quad (34)$$

Moreover, to reject biologically unreachable equilibrium states, it is necessary to preserve the condition.

$$h_S + h_R - 2c_R \geq 0. \quad (35)$$

As Pimenov et al. argue [50], this condition implies that prey in both modes, risky and safe, is taking advance of the same environment and using the same resources. This basically differs from the concepts of multiple patches or refuge, when the prey in these habitats is assumed to have no access to resources in other patches, and hence all $h_{ij} = 0$.

Considering again that the system dynamics are unknown, an extensive set of input-output data have been obtained from the state model in (34). Subsequently, a batch of different structures of MLP neural networks has been trained in order to extract an appropriate NN that accurately reproduces the behaviour of the predator-prey system. This dataset contains 10,000 points uniformly extracted from the universes of discourse of the state variables, $x_1 = p(t)$ and $x_2 = q(t)$, and input, $u(t) = K_R(t)$, respectively.

$$\begin{aligned}X &\subseteq \{\mathbf{x} \mid x_1(t) \in [0, 30], x_2(t) \in [0, 70]\}, \\ U &\subseteq \{u \mid u(t) \in [5, 100]\},\end{aligned}\quad (36)$$

where 8000 data (80%) have been used for training MLP neural networks, proceeding in previous examples, and 2000 data (20%) have been managed in the validation stage to select an appropriate neural model. After training and validating, for each of the state variables, a MLP structure

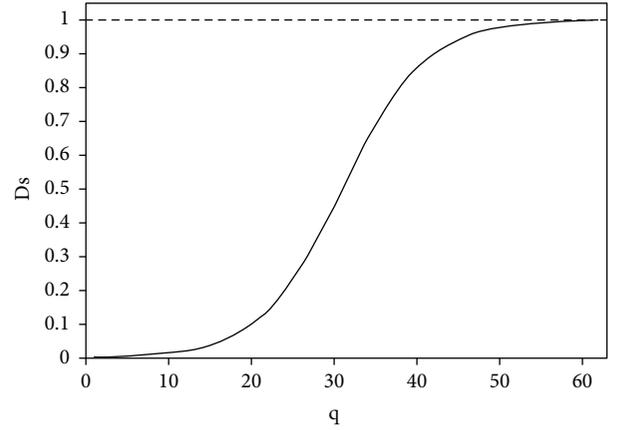


FIGURE 6: Functional response: probability of hunted prey $Ds(q)$ versus size of predator population q .

with a different dimension has been selected, incorporating 15 neurons in the hidden layer for the estimation of \dot{x}_1 and 6 neurons in the case of \dot{x}_2 . The training process has been performed with the same algorithm as above, the second order L-M algorithm [44, 47], obtaining a validation MSE result of 17.797 and 0.1059 individuals, respectively.

Afterwards, the set of equations in (23) has been solved with the same approach as before, the L-M algorithm adapted by Moré [44, 45], through a dense mesh of points created into the universes of discourse of the studied variables, $p(t)$, $q(t)$, and $K_R(t)$, applying increments of 1 individual. Figure 7 shows the comparison between the solutions obtained through the NN model and the real system.

After the linearization developed in this work, different equilibrium states have been found in the real system, as shown in Figure 7. These states are characterized by their eigenvalues of the corresponding dynamic matrix. In the same way, it can be seen how the equilibrium states calculated by the NN model have correlation with the states existing in the real system. All these results are directly related to the analysis of the real system that Pimenov shows at work [50], and they maintain a precise correspondence with them. As in previous examples, those points calculated outside of the universe of discourse can also be discarded.

5. Conclusions

This work has presented a straightforward methodological procedure based on neural network models, to analyze the dynamical behaviour of unknown nonlinear systems. This implies presenting the linearization of a neural model to be compared with the proper linearization of the real system, and finally explaining the calculation of the equilibrium states of the real system through the NN model and its relation with the study of the local stability.

This work initiates a research line where NNs are the basis of a nonlinear system model, allowing to calculate its corresponding Jacobian matrix to analyze the local stability of the equilibrium states. Since it is assumed that there is no approximate mathematical model of the nonlinear

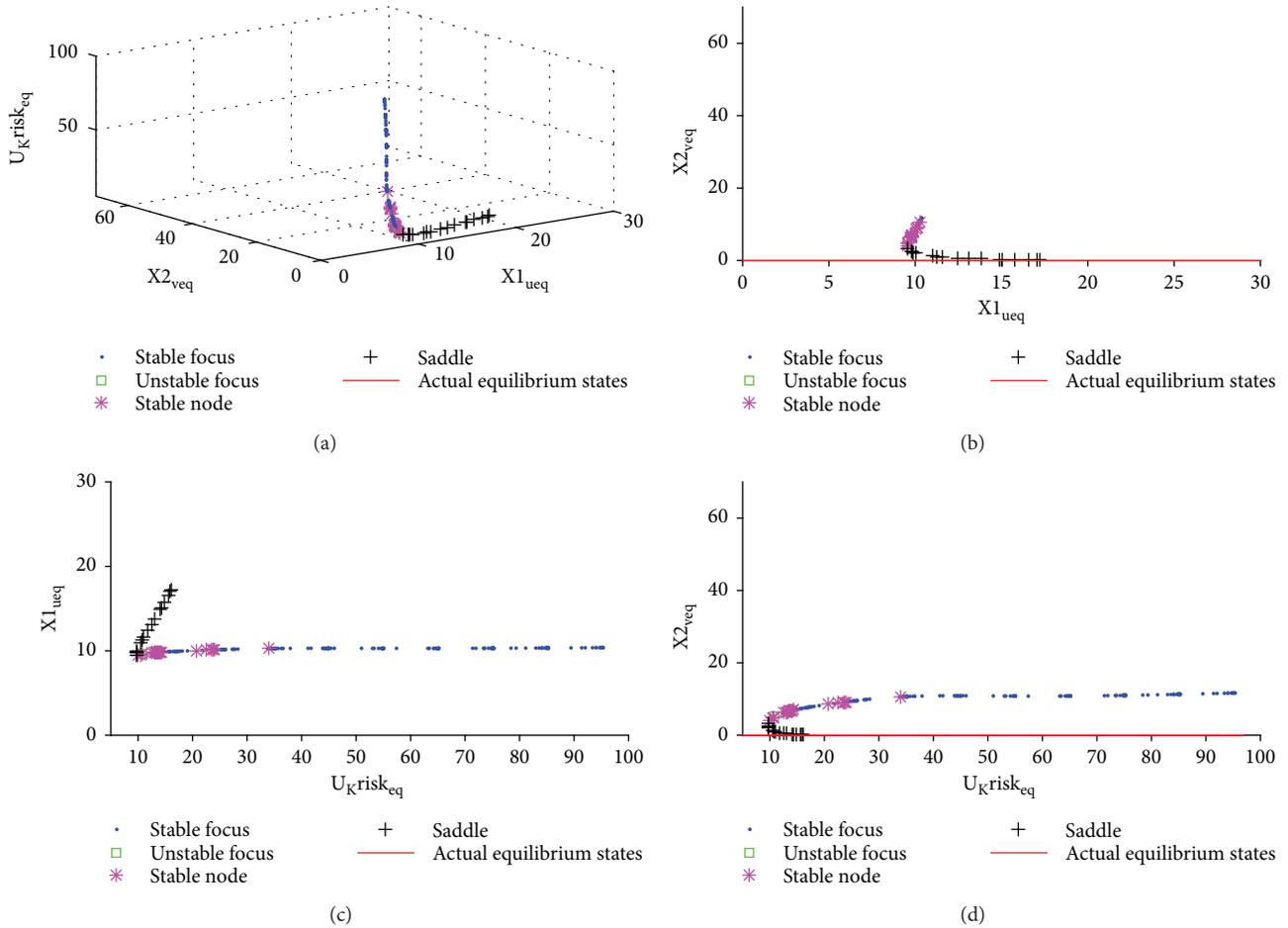


FIGURE 7: Equilibrium states of the predator-prey NN model.

system, the accuracy of the study of equilibrium states is related to the NN training process. In this process, many experiments have been developed to ensure the representativeness of the entire operating range that will be considered in the analysis of the nonlinear system. This has involved keeping the system under control and acquiring the appropriate amount of examples of variables and states.

The proposed method has been validated with three nonlinear systems coming from different fields (electrical, mechanical, and biological). This approach has been developed with multilayer perceptron neural network structures, one of the most simple and feasible in training and application. The training processes have been based on widely used Levenberg-Marquardt algorithm and developed with exhaustive input-output data of the above systems, guaranteeing the representativeness of each universe of discourse. Taking into account that after the training the errors have been within permissible limits, it is necessary to emphasize that the extracted information, with this neural approach, through the analysis of the neural structure is a powerful method to obtain information from an unknown system.

The results have shown that it is possible to obtain dynamical information of nonlinear systems uniquely using the corresponding NN model. This information is directly

related to the equilibrium states of the real system, enabling to study the local stability around each equilibrium state.

Due to the interest of this research line, more additional works could be developed in the future considering other NN structures different to that presented in this paper, the multilayer perceptron with sigmoidal activation function in its unique hidden layer. The study of nonlinear systems of more complicated dynamic features (including systems with memory and with noncontinuous operating ranges), together with the modification of the NN structures (recurrent, radial basis function, deep neural networks, etc.), could establish a promising and extensive field of study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank the Ministry of Economy, Industry and Competitiveness of Spain that has funded this work under the project DPI2017-85540-R (H2SMART-GRID).

References

- [1] A. Zubizarreta, M. Larrea, E. Irigoyen, I. Cabanes, and E. Portillo, "Real time direct kinematic problem computation of the 3prs robot using neural networks," *Neurocomputing*, vol. 271, Supplement C, pp. 104–114, 2018.
- [2] J. M. Andújar and J. M. Bravo, "Multivariable fuzzy control applied to the physical–chemical treatment facility of a cellulose factory," *Fuzzy Sets and Systems*, vol. 150, no. 3, pp. 475–492, 2005.
- [3] M. E. Ortiz-Quisbert, M. A. Duarte-Mermoud, F. Milla, and R. Castro-Linares, "Control Adaptativo Fraccionario Optimizado por Algoritmos Genéticos, Aplicado a Reguladores Automáticos de Voltaje," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 13, no. 4, pp. 403–409, 2016.
- [4] E. Irigoyen, M. Larrea, J. Valera, V. Gómez, and F. Artaza, "A hybridized neurogenetic solution for controlling industrial r3 workspace," *Neural Network World*, vol. 20, no. 7, pp. 811–824, 2010.
- [5] J. G. G. Fontanet, A. L. Cervantes, and I. B. Ortiz, "Alternativas de control para un Péndulo de Furuta," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 13, no. 4, pp. 410–420, 2016.
- [6] E. Irigoyen and G. Miñano, "A narx neural network model for enhancing cardiovascular rehabilitation therapies," *Neurocomputing*, vol. 109, pp. 9–15, 2013.
- [7] H. K. Khalil, *Nonlinear Systems*, Prentice-Hall, Upper Saddle River, NJ, USA, 2000.
- [8] W. J. Rugh, *Linear System Theory*, Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1996.
- [9] T. Jing and F. Chen, "Finite-time lag synchronization of delayed neural networks via periodically intermittent control," *Complexity*, vol. 21, no. S1, pp. 211–219, 2016.
- [10] S. Lei and H. Shen, "Fault-tolerant mixed \mathcal{H}_∞ /passive synchronization for delayed chaotic neural networks with sampled-data control," *Complexity*, vol. 21, no. 6, pp. 246–259, 2016.
- [11] J. A. Grande, J. M. Andújar, J. Aroba et al., "Fuzzy modeling of the spatial evolution of the chemistry in the tinto river (SW Spain)," *Water Resources Management*, vol. 24, no. 12, pp. 3219–3235, 2010.
- [12] M. Larrea, E. Irigoyen, V. Gómez, and F. Artaza, "Nonlinear system control based on neural networks with adaptive predictive strategy," in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, pp. 1–7, Bilbao, Spain, September 2010.
- [13] M. J. López-Baldán, A. García-Cerezo, J. M. C. López, and A. R. Gallego, "Fuzzy modeling of a thermal solar plant," *International Journal of Intelligent Systems*, vol. 17, no. 4, pp. 369–379, 2002.
- [14] S. Chen and S. A. Billings, "Neural networks for nonlinear dynamic system modelling and identification," *International Journal of Control*, vol. 56, no. 2, pp. 319–346, 1992.
- [15] J. J. V. García, V. G. Garay, E. I. Gordo, F. A. Fano, and M. L. Sukia, "Intelligent multi-objective nonlinear model predictive control (imo-nmpc): towards the 'on-line' optimization of highly complex control problems," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6527–6540, 2012.
- [16] H. J. Marquez, *Nonlinear Control Systems. Analysis and Design*, John Wiley & Sons, Inc., 2003.
- [17] J. M. Andújar, J. M. Bravo, and A. Peregrín, "Stability analysis and synthesis of multivariable fuzzy systems using interval arithmetic," *Fuzzy Sets and Systems*, vol. 148, no. 3, pp. 337–353, 2004.
- [18] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos. Texts in Applied Mathematics*, Springer, 2nd edition, 2003.
- [19] S. Sastry, *Nonlinear System: Analysis, Stability, and Control*, Springer, New York, NY, USA, 1999.
- [20] A. J. Barragán, B. M. Al-Hadithi, A. Jiménez, and J. M. Andújar, "A general methodology for online TS fuzzy modeling by the extended kalman filter," *Applied Soft Computing*, vol. 18, pp. 277–289, 2014.
- [21] A. J. Barragán, B. M. Al-Hadithi, J. M. Andújar, and A. Jiménez, "Metodología formal de análisis del comportamiento dinámico de sistemas no lineales mediante lógica borrosa," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 12, no. 4, pp. 434–445, 2015.
- [22] A. Sahoo, H. Xu, and S. Jagannathan, "Neural network-based event-triggered state feedback control of nonlinear continuous-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 3, pp. 497–509, 2016.
- [23] K. Pirabakaran and V. M. Becerra, "Pid autotuning using neural networks and model reference adaptive control," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 451–456, 2002.
- [24] N. Kishor, R. P. Saini, and S. P. Singh, "Small hydro power plant identification using nnarx structure," *Neural Computing and Applications*, vol. 14, no. 3, pp. 212–222, 2005.
- [25] R. K. Al Seyab and Y. Cao, "Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation," *Journal of Process Control*, vol. 18, no. 6, pp. 568–581, 2008.
- [26] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [27] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [28] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.
- [29] V. Z. Marmarelis and X. Zhao, "Volterra models and three-layer perceptrons," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1421–1433, 1997.
- [30] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, no. 5, pp. 551–560, 1990.
- [31] A. R. Gallant and H. White, "On learning the derivatives of an unknown mapping with multilayer feedforward networks," *Neural Networks*, vol. 5, no. 1, pp. 129–138, 1992.
- [32] J. M. A. Marquez, A. J. B. Pina, and M. E. G. Arias, "A general and formal methodology to design stable nonlinear fuzzy control systems," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1081–1091, 2009.
- [33] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [34] S. Jagannathan, F. L. Lewis, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, 1999.

- [35] N. K. Poulsen, M. Nørgaard, O. Ravn, and L. K. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook*, Springer, 2000.
- [36] S. Jagannathan, *Neural Network Control of Nonlinear Discrete-Time Systems*, Francis & Taylor, 2006.
- [37] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer Verlag, Berlin, Germany, 1990.
- [38] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Prentice-Hall, Upper Saddle River, NJ, USA, 1991.
- [39] M. Larrea, *Estrategia Adaptativa-Predictiva Basada en Redes Neuronales para el Control de Sistemas No Lineales*, [Ph.D. thesis], UPV/EHU, 2014, Editor: CreateSpace Independent Publishing Platform.
- [40] Q. Song, Y. Wu, and Y. C. Soh, "Robust adaptive gradient-descent training algorithm for recurrent neural networks in discrete time domain," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1841–1853, 2008.
- [41] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor uav using neural networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 1, pp. 50–66, 2010.
- [42] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter search and local NLP solvers: a multistart framework for global optimization," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.
- [43] F. Glover, "A template for scatter search and path relinking," in *Artificial Evolution. AE 1997*, vol. 1363 of Lecture Notes in Computer Science, pp. 1–51, Springer, Berlin Heidelberg, 1998.
- [44] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [45] J. J. Moré G. A. Watson, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical Analysis*, pp. 105–116, Springer, Berlin, Heidelberg, 1978.
- [46] E. Irigoyen and M. Pinzolas, "Numerical bounds to assure initial local stability of narx multilayer perceptrons and radial basis functions," *Neurocomputing*, vol. 72, no. 1–3, pp. 539–547, 2008.
- [47] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.
- [48] E. I. Butikov, "On the dynamic stabilization of an inverted pendulum," *American Journal of Physics*, vol. 69, no. 7, pp. 755–768, 2001.
- [49] A. Korobeinikov, "Stability of ecosystem: global properties of a general predator-prey model," *Mathematical Medicine and Biology*, vol. 26, no. 4, pp. 309–321, 2009.
- [50] A. Pimenov, T. C. Kelly, A. Korobeinikov, M. J. A. O'Callaghan, and D. Rachinskii, "Adaptive behaviour and multiple equilibrium states in a predator-prey model," *Theoretical Population Biology*, vol. 101, pp. 24–30, 2015.

Research Article

Some Hesitant Fuzzy Linguistic Muirhead Means with Their Application to Multiattribute Group Decision-Making

Jun Wang ¹, Runtong Zhang ¹, Xiaomin Zhu ², Yuping Xing¹ and Borut Buchmeister³

¹School of Economics and Management, Beijing Jiaotong University, Beijing 100044, China

²School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China

³Faculty of Mechanical Engineering, University of Maribor, 2000 Maribor, Slovenia

Correspondence should be addressed to Runtong Zhang; rtzhang@bjtu.edu.cn

Received 5 September 2017; Revised 12 April 2018; Accepted 3 May 2018; Published 26 June 2018

Academic Editor: Jesus Medina

Copyright © 2018 Jun 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.

The proposed hesitant fuzzy linguistic set (HFLS) is a powerful tool for expressing fuzziness and uncertainty in multiattribute group decision-making (MAGDM). This paper aims to propose novel aggregation operators to fuse hesitant fuzzy linguistic information. First, we briefly recall the notion of HFLS and propose new operations for hesitant fuzzy linguistic elements (HFLEs). Second, considering the Muirhead mean (MM) is a useful aggregation technology that can consider the interrelationship among all aggregated arguments, we extend it to hesitant fuzzy linguistic environment and propose new hesitant fuzzy linguistic aggregation operators, such as the hesitant fuzzy linguistic Muirhead mean (HFLMM) operator, the hesitant fuzzy linguistic dual Muirhead mean (HFLDMM) operator, the hesitant fuzzy linguistic weighted Muirhead mean (HFLMM) operator, and the hesitant fuzzy linguistic weighted dual Muirhead mean (HFLWDMM) operator. These operators can reflect the correlations among all HFLEs. Several desirable properties and special cases of the proposed operators are also studied. Furthermore, we propose a novel approach to MAGDM in a hesitant fuzzy linguistic context based on the proposed operators. Finally, we conduct a numerical experiment to demonstrate the validity of our method. Additionally, we compare our method with others to illustrate its merits and superiorities.

1. Introduction

MAGDM is an activity that selects the optimal alternative under a set of attributes assessed by a group of decision-makers. Owing to the increased complexity in decision-making, one of the difficulties in practical MAGDM problems is representing attribute values in fuzzy and vague decision-making environments. In 1965, Zadeh [1] originally proposed an effective tool, called fuzzy set (FS), for depicting and expressing impreciseness and uncertainty. Since its introduction, FS has received substantial attention and has been studied by thousands of scientists worldwide in theoretical and practical aspects [2]. Thereafter, several extensions of FS have been proposed, such as interval-valued fuzzy set [3], intuitionistic fuzzy set (IFS) [4], interval-valued intuitionistic fuzzy set [5], type 2 fuzzy set [6], and neutrosophic set [7]. In the past decades, these fuzzy sets have been successfully applied to decision-making [8–26]. However, these tools

are unsuitable to cope with circumstances in which decision-makers are hesitant between a few different values when determining membership degree. Therefore, Torra [27] proposed the concept of hesitant fuzzy set (HFS), which permits the membership degree of an element to a set to be represented by a set of possible values between 0 and 1, in order to address such cases.

Additionally, decision-makers cannot make quantitative decisions with limited priori knowledge and insufficient time due to the high complicity of actual decision-making problems. Thus, qualitative methods are used to express the decision-makers' preference information. The linguistic term set (LTS) can be used for the convenient assessment of linguistic variables rather than numerical values. Motivated by the HFS, Rodríguez et al. [28] proposed the hesitant fuzzy linguistic term set (HFLTS), in which several linguistic terms are used to evaluate a linguistic variable. However, the HFLTS cannot reflect possible membership degrees of a

linguistic term to a given concept [29]. For example, in an intuitionistic linguistic set [30], a linguistic variable and an intuitionistic fuzzy number are used to describe the fuzzy attributes of an alternative. To overcome the drawback of HFLTSS, Lin et al. [31] introduced HFLSSs, in which HFSs are utilized to express the hesitancy of decision-makers in selecting the membership degrees for a linguistic term. The basic elements of the HFLSSs are called HFLEs. For example, a possible HFLE can be denoted as $a = \langle s_2, \{0.1, 0.3, 0.7\} \rangle$, where s_2 is a linguistic term, and $\{0.1, 0.3, 0.7\}$ is a collection of membership degrees that describe fuzziness, uncertainty, and hesitancy of decision-makers when providing a linguistic term. Evidently, $\{0.1, 0.3, 0.7\}$ is a hesitant fuzzy element (HFE). HFLS can be employed to evaluate an object from two aspects, namely, a linguistic term and an HFE. The former can evaluate the object as “medium,” “poor,” or “too poor,” and the latter can express the hesitancy of decision-makers in giving a linguistic term.

In the MAGDM process, one of the most significant steps is aggregating the decision-makers' preference information. In the past years, aggregation operators have gained increasing research attention. One of the most classical and popular aggregation operators is the ordered weighted averaging (OWA) operator, which was introduced by Yager [32] for crisp numbers. The OWA operator was extended to IFSs [33], HFSs [34], and dual HFSs [35]. Moreover, researchers have proposed extensions of the OWA operators, such as induced OWA [36] and generalized OWA operator [37]. However, the OWA operator and its extensions do not consider the interrelationship among the arguments. Thus, these operators assume that attributes are independent, which is inconsistent with reality. Therefore, scholars have focused on operators which can capture the interrelationship among arguments. The Bonferroni mean (BM) [38] and Heronian mean (HM) [39] are two crucial aggregation technologies that consider the interrelationship between any two arguments. Recently, these operators have been extended to aggregate hesitant fuzzy linguistic information, and a number of hesitant fuzzy linguistic aggregation operators have been proposed [31, 40–43]. However, the correlations among arguments are ubiquitous, which means that the interrelationship among all arguments should be considered, as such, the BM and HM are inadequate and insufficient. The MM [44] is a well-known aggregation operator that considers the interrelationship among all arguments and possesses a parameter vector that leads to a flexible aggregation process. In addition, some existing operators are special cases of MM. Recently, MM has been investigated in intuitionistic fuzzy [45] and 2-tuple linguistic environments [46]. To the best of our knowledge, no research has been performed on MM under HFLSSs. Hence, the MM operator should be extended to HFLSSs. The present study investigates the MM under hesitant fuzzy linguistic environment and proposes new aggregation operators for HFLEs. The contribution of this paper is that we propose new operators for aggregating hesitant fuzzy linguistic information that can capture the interrelationship among all HFLEs. Furthermore, we apply the developed operators to MAGDM in which attribute values take the form of HFLEs.

The aims and motivations of this paper are (1) to propose some new aggregation operators to aggregate HFLEs and (2) to propose a novel approach to MAGDM problems. The rest of the paper is organized as follows. Section 2 briefly recalls basic concepts, such as HFLS and MM. Section 3 proposes several hesitant fuzzy linguistic Muirhead mean operators. Section 4 describes the developed weighted aggregation operators. Section 5 proposes a novel approach to MAGDM within the hesitant fuzzy linguistic context. Section 6 validates the proposed method by providing a numerical example. The final section summarizes the paper.

2. Preliminaries

In this section, we briefly review concepts about HFLSSs and their operations. The concepts of MM and dual Muirhead mean (DMM) are also introduced.

2.1. Linguistic Term Sets and Hesitant Fuzzy Linguistic Sets

Definition 1. Let $S = \{s_i \mid i = 1, 2, \dots, t\}$ be an LTS with odd cardinality, where s_i represents the possible value for linguistic variable and should satisfy the following [47]: (1) the set is ordered: $s_i > s_j$ if $i > j$; (2) the negation operator $\text{neg}(s_i) = s_j$ is set, such that $i + j = t + 1$; (3) the max operator is $\max(s_i, s_j) = s_i$ if $s_i \leq s_j$; and (4) the min operator is $\min(s_i, s_j) = s_i$ if $s_i \leq s_j$. For example, when $t = 7$, a possible LTS S can be defined as

$$\begin{aligned} S &= \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\} \\ &= \{\text{very poor, poor, slightly poor, fair,} \\ &\quad \text{slightly good, good, very good}\}. \end{aligned} \quad (1)$$

Lin et al. [31] then introduced HFLSSs based on HFSs and LTSs.

Definition 2 [31]. Let X be an ordinary fixed set, then a hesitant fuzzy linguistic set (HFLS) A on X can be defined as

$$A = \left\langle \left\langle x, s_{\theta(x)}, h_A(x) \right\rangle \mid x \in X \right\rangle, \quad (2)$$

where $s_{\theta(x)}$ is the linguistic term, and $h_A(x)$ is the HFE that denotes the possible membership degrees of the element $x \in X$ to $s_{\theta(x)}$. For convenience, $a = \langle s_{\theta(x)}, h_A(x) \rangle$ is called an HFLE by Lin et al. [31].

Example 1. Let $X = \{x_1, x_2, x_3\}$ be an ordinary fixed set. A possible HFLS A defined on X can be $A = \{ \langle x, s_{\theta(x)}, h_A(x) \rangle \mid x \in X \} = \{ \langle x_1, s_5, \{0.3\} \rangle, \langle x_2, s_4, \{0.1, 0.6\} \rangle, \langle x_3, s_4, \{0.2, 0.4, 0.5\} \rangle \}$. If we divide A into three subsets that contain only one object, then the three HFLEs are $\langle s_5, \{0.3\} \rangle$, $\langle s_4, \{0.1, 0.6\} \rangle$, and $\langle s_6, \{0.2, 0.4, 0.5\} \rangle$. In $\langle s_5, \{0.3\} \rangle$, 0.3 denotes the membership degree that x_1 belongs to s_5 . In $\langle s_4, \{0.1, 0.6\} \rangle$, 0.1 and 0.6 denote the possible membership degrees that x_2 belongs to s_4 . In $\langle s_6, \{0.2, 0.4, 0.5\} \rangle$, 0.2, 0.4, and 0.5 represent the possible membership degrees that x_3 belongs to s_6 . Notably, $\langle s_5, \{0.3\} \rangle$ is a special case of HFLE in which only one membership degree is assigned in the corresponding HFE.

Evidently, an HFLE is a combination of linguistic terms with HFE, which takes the advantages of the two. Compared with linguistic terms, HFLE contains an HFE that permits several possible membership degrees and denotes the degrees to which an alternative belongs to in a corresponding linguistic term. Therefore, HFLE can more accurately and appropriately express the fuzziness, uncertainty, and hesitancy of decision-makers than crisp linguistic variables. Compared with HFE, HFLE has a linguistic term that evaluates an object as “poor,” “middle,” or “good.” Hence, linguistic terms and HFEs can only evaluate objects from one aspect, whereas HFLSs can evaluate objects from two aspects, namely, qualitative (linguistic terms) and quantitative evaluations (HFEs). Therefore, HFLSs are more powerful than HFSs and LTSs.

HFLEs can be used in real decision-making problems. For example, the linguistic term (s_2) “poor” is acceptable for evaluating the functionality and technology of an ERP system by four decision-makers. Three decision-makers are required to provide their preference information under the value “poor” (s_2) . If the first decision-maker provides 0.1, the second decision-maker provides 0.3, the third decision-maker provides 0.6, and the fourth decision-maker provides 0.8, then the combination evaluation can be denoted by $\langle s_2, \{0.1, 0.3, 0.6, 0.8\} \rangle$. Based on this analysis, HFLS is a powerful and effective decision-making tool.

Lin et al. [31] introduced a law to compare any two HFLEs.

Definition 3 [31]. For an HFLE $a = \langle s_{\theta(a)}, h(a) \rangle$, the score function of a is $S(a) = ((1/\#h) \sum_{\gamma \in h} \gamma) s_{\theta(a)}$, where $\#h$ is the number of values in h ; for convenience, $\#h$ is also called the length of h . For any two HFLEs a_1 and a_2 , if $s(a_1) > s(a_2)$, then $a_1 > a_2$; if $s(a_1) = s(a_2)$, then $a_1 = a_2$.

Additionally, Lin et al. [31] introduced several operations for HFLEs.

Definition 4 [31]. Let $a = \langle s_{\theta(a)}, h(a) \rangle$, $a_1 = \langle s_{\theta(a_1)}, h(a_1) \rangle$, and $a_2 = \langle s_{\theta(a_2)}, h(a_2) \rangle$ be any three HFLEs, and λ be a positive crisp number, then

- (1) $a_1 \oplus a_2 = \langle s_{\theta(a_1)} + s_{\theta(a_2)}, \cup_{\gamma(a_1) \in h(a_1), \gamma(a_2) \in h(a_2)} \{\gamma(a_1) + \gamma(a_2) - \gamma(a_1)\gamma(a_2)\} \rangle$,
- (2) $a_1 \otimes a_2 = \langle s_{\theta(a_1)} \times s_{\theta(a_2)}, \cup_{\gamma(a_1) \in h(a_1), \gamma(a_2) \in h(a_2)} \{\gamma(a_1)\gamma(a_2)\} \rangle$,
- (3) $a^\lambda = \langle s_{\theta(a)}^\lambda, \cup_{\gamma(a) \in h(a)} \{\gamma(a)^\lambda\} \rangle$,
- (4) $\lambda a = \langle \lambda s_{\theta(a)}, \cup_{\gamma(a) \in h(a)} \{1 - (1 - \gamma(a))^\lambda\} \rangle$.

However, these operations for HFLEs are complicated to use. For example, let $a = a_1 \otimes a_2$, by Definition 4, we obtain $\#h(a) = (\#h(a_1)) \times (\#h(a_2))$. When aggregating a set of HFLEs, the aggregated values are very complicated. Hence, we should simplify the operations for HFLEs. Motivated by the simplified operations for HFEs introduced by Liao et al. [48], we introduce new operations for HFLEs.

Definition 5. Let $a = \langle s_{\theta(a)}, h(a) \rangle$, $a_1 = \langle s_{\theta(a_1)}, h(a_1) \rangle$, and $a_2 = \langle s_{\theta(a_2)}, h(a_2) \rangle$ be any of the three HFLEs satisfying $\#h(a) = \#h(a_1) = \#h(a_2)$ and λ be a positive crisp number, then

- (1) $a_1 \oplus a_2 = \langle s_{\theta(1)} + s_{\theta(2)}, \{h_1^{\sigma(t)} + h_2^{\sigma(t)} - h_1^{\sigma(t)}h_2^{\sigma(t)}\} \rangle$,
- (2) $a_1 \otimes a_2 = \langle s_{\theta(a_1)} \times s_{\theta(a_2)}, \{h_1^{\sigma(t)}h_2^{\sigma(t)}\} \rangle$,
- (3) $a^\lambda = \langle s_{\theta(a)}^\lambda, \{(h^{\sigma(t)})^\lambda\} \rangle$,
- (4) $\lambda a = \langle \lambda s_{\theta(a)}, \{1 - (1 - h^{\sigma(t)})^\lambda\} \rangle$, where $t = 1, 2, \dots, \#h(a)$, $h^{\sigma(t)}$, $h_1^{\sigma(t)}$, and $h_2^{\sigma(t)}$ represent the t^{th} smallest values of $h(a)$, $h(a_1)$, and $h(a_2)$, respectively.

Remark 1. In Definition 5, we assume that all HFEs have the same number of values that cannot be always satisfied. To solve the problem, Xu and Xia [49] introduced a transformation regulation for HFEs by assuming that all decision-makers are pessimistic (or optimistic). The transformation regulation can be described as follows. For two HFEs h_1 and h_2 , let $h = \max\{\#h_1, \#h_2\}$. If $\#h_1 < \#h_2$, then h_1 should be extended by adding the minimum (or maximum) value until it has the number of values as h_2 . If $\#h_1 < \#h_2$, then h_2 should be extended by adding the minimum (or maximum) value until it has the same number of values as h_1 .

2.2. Muirhead Mean. MM is an aggregation operator for crisp numbers introduced by Muirhead [44]. This operator can capture the interrelationship among all aggregated arguments.

Definition 6 [44]. Let $a_i (i = 1, 2, \dots, n)$ be a collection of crisp numbers and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then MM is defined as

$$\text{MM}^P(a_1, a_2, \dots, a_n) = \left(\frac{1}{n!} \sum_{\vartheta \in S_n} \prod_{j=1}^n a_{\vartheta(j)}^{p_j} \right) \frac{1}{\sum_{j=1}^n p_j}, \quad (3)$$

Where $\vartheta(j) (j = 1, 2, \dots, n)$ is any permutation of $(1, 2, \dots, n)$, and S_n is the collection of all permutations of $(1, 2, \dots, n)$.

Furthermore, Liu and Li [45] proposed the DMM operator.

Definition 7 [45]. Let $a_i (i = 1, 2, \dots, n)$ be a collection of crisp numbers and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then the DMM operator is defined as

$$\text{DMM}^P(a_1, a_2, \dots, a_n) = \frac{1}{\sum_{j=1}^n p_j} \left(\prod_{\vartheta \in S_n} \prod_{j=1}^n (p_j a_{\vartheta(j)}) \right)^{1/n!}, \quad (4)$$

where $\vartheta(j) (j = 1, 2, \dots, n)$ is any permutation of $(1, 2, \dots, n)$, and S_n is the collection of all permutations of $(1, 2, \dots, n)$.

3. Hesitant Fuzzy Linguistic Muirhead Mean Operators

In this section, we extend MM and DMM to the hesitant fuzzy linguistic environment and develop new aggregation operators for aggregating HFLEs.

3.1. Hesitant Fuzzy Linguistic Muirhead Mean Operator

Definition 8. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs where $\#h(a_i) = h$ holds for all i , and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then the hesitant fuzzy linguistic Muirhead mean (HFLMM) can be defined as

$$\text{HFLMM}^P(a_1, a_2, \dots, a_n) = \left(\frac{1}{n!} \bigoplus_{\vartheta \in S_n} \left(\bigotimes_{j=1}^n a_{\vartheta(j)}^{p_j} \right) \right)^{1/\sum_{j=1}^n p_j}, \quad (5)$$

where $\vartheta(j) (j = 1, 2, \dots, n)$ is any permutation of $(1, 2, \dots, n)$, and S_n is the collection of all permutations of $(1, 2, \dots, n)$.

According to the operations for HFLEs, the following theorem can be obtained.

Theorem 1. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs with $\#h(a_i) = h$ holding for all i and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then the aggregated value of HFLMM is also an HFLE and

$$\text{HFLMM}^P(a_1, a_2, \dots, a_n) = \left\langle \left(\frac{1}{n!} \sum_{\vartheta \in S_n} \prod_{j=1}^n (s_{\theta_{\vartheta(j)}})^{p_j} \right)^{1/\sum_{j=1}^n p_j}, \left\{ \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n (h_{\theta_{\vartheta(j)}}^{\sigma(t)})^{p_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n p_j} \mid t = 1, 2, \dots, h \right\} \right\rangle. \quad (6)$$

Proof 1. According to Definition 5, we have

$$a_{\vartheta(j)}^{p_j} = \left\langle (s_{\theta_{\vartheta(j)}})^{p_j}, \left\{ (h_{\theta_{\vartheta(j)}}^{\sigma(t)})^{p_j} \mid t = 1, 2, \dots, h \right\} \right\rangle, \quad \bigotimes_{j=1}^n a_{\vartheta(j)}^{p_j} = \left\langle \prod_{j=1}^n (s_{\theta_{\vartheta(j)}})^{p_j}, \left\{ \prod_{j=1}^n (h_{\theta_{\vartheta(j)}}^{\sigma(t)})^{p_j} \mid t = 1, 2, \dots, h \right\} \right\rangle. \quad (7)$$

Thus,

$$\bigoplus_{\vartheta \in S_n} \left(\bigotimes_{j=1}^n a_{\vartheta(j)}^{p_j} \right) = \left\langle \sum_{\vartheta \in S_n} \prod_{j=1}^n (s_{\theta_{\vartheta(j)}})^{p_j}, \left\{ 1 - \prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n (h_{\theta_{\vartheta(j)}}^{\sigma(t)})^{p_j} \mid t = 1, 2, \dots, h \right) \right\} \right\rangle. \quad (8)$$

Furthermore,

$$\frac{1}{n!} \bigoplus_{\vartheta \in S_n} \left(\bigotimes_{j=1}^n a_{\vartheta(j)}^{p_j} \right) = \left\langle \frac{1}{n!} \sum_{\vartheta \in S_n} \prod_{j=1}^n (s_{\theta_{\vartheta(j)}})^{p_j}, \left\{ 1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n (h_{\theta_{\vartheta(j)}}^{\sigma(t)})^{p_j} \right) \right)^{1/n!} \mid t = 1, 2, \dots, h \right\} \right\rangle. \quad (9)$$

Therefore

$$\left(\frac{1}{n!} \bigoplus_{\vartheta \in S_n} \left(\bigotimes_{j=1}^n a_{\vartheta(j)}^{p_j} \right) \right)^{1/\sum_{j=1}^n p_j} = \left\langle \left(\frac{1}{n!} \sum_{\vartheta \in S_n} \prod_{j=1}^n (s_{\theta_{\vartheta(j)}})^{p_j} \right)^{1/\sum_{j=1}^n p_j}, \left\{ \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n (h_{\theta_{\vartheta(j)}}^{\sigma(t)})^{p_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n p_j} \mid t = 1, 2, \dots, h \right\} \right\rangle. \quad (10)$$

Hence, (6) is maintained.

Considering that $h_{\vartheta(j)}^{\sigma(t)} \in [0, 1]$, we can obtain

$$\begin{aligned} & \left(h_{\vartheta(j)}^{\sigma(t)} \right)^{P_j} \in [0, 1], \prod_{j=1}^n \left(h_{\vartheta(j)}^{\sigma(t)} \right)^{P_j} \\ & \in [0, 1], 1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \in [0, 1]. \end{aligned} \quad (11)$$

Therefore,

$$\begin{aligned} & \prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n \left(h_{\vartheta(j)}^{\sigma(t)} \right)^{P_j} \right) \\ & \in [0, 1], \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n \left(h_{\vartheta(j)}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!} \in [0, 1]. \end{aligned} \quad (12)$$

Thus,

$$1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n \left(h_{\vartheta(j)}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!} \in [0, 1],$$

$$\cdot \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n \left(h_{\vartheta(j)}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n P_j} \in [0, 1], \quad (13)$$

where $t = 1, 2, \dots, h$. Therefore, (6) is an HFLE that completes the proof.

The HFLMM operator has the following properties.

Theorem 2 (monotonicity). *Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ and $b_i = \langle s_{\theta(b_i)}, h(b_i) \rangle (i = 1, 2, \dots, n)$ be two collections of HFLEs; if these conditions are satisfied for all i : (1) $\#h(a_i) = \#h(b_i) = h$; (2) $s_{\theta(a_i)} \leq s_{\theta(b_i)}$; (3) $h_{a_i}^{\sigma(t)} \leq h_{b_i}^{\sigma(t)}$, where $t = 1, 2, \dots, h$, then*

$$\text{HFLMM}^P(a_1, a_2, \dots, a_n) \leq \text{HFLMM}^P(b_1, b_2, \dots, b_n). \quad (14)$$

Proof 2. Let $\text{HFLMM}^P(a_1, a_2, \dots, a_n) = a = \langle s_{\theta(a)}, h(a) \rangle$ and $\text{HFLMM}^P(b_1, b_2, \dots, b_n) = b = \langle s_{\theta(b)}, h(b) \rangle$.

From Theorem 1, we can know that $a = \langle s_{\theta(a)}, h(a) \rangle$ and $b = \langle s_{\theta(b)}, h(b) \rangle$ are two HFLEs and

$$\begin{aligned} s_{\theta(a)} &= \left(\frac{1}{n!} \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(a_{i_j})} \right)^{P_j} \right)^{1/\sum_{j=1}^n P_j}, h(a) = \left\{ \left(1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n P_j} \mid t = 1, 2, \dots, h \right\}, \\ s_{\theta(b)} &= \left(\frac{1}{n!} \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(b_{i_j})} \right)^{P_j} \right)^{1/\sum_{j=1}^n P_j}, h(b) = \left\{ \left(1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{b_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n P_j} \mid t = 1, 2, \dots, h \right\}. \end{aligned} \quad (15)$$

Given that $s_{\theta(a_i)} \leq s_{\theta(b_i)}$, we have

$$\left(s_{\theta(a_{i_j})} \right)^{P_j} \leq \left(s_{\theta(b_{i_j})} \right)^{P_j}, \prod_{j=1}^n \left(s_{\theta(a_{i_j})} \right)^{P_j} \leq \prod_{j=1}^n \left(s_{\theta(b_{i_j})} \right)^{P_j}. \quad (16)$$

Therefore,

$$\begin{aligned} & \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(a_{i_j})} \right)^{P_j} \leq \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(b_{i_j})} \right)^{P_j}, \\ & \frac{1}{n!} \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(a_{i_j})} \right)^{P_j} \leq \frac{1}{n!} \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(b_{i_j})} \right)^{P_j}. \end{aligned} \quad (17)$$

Thus,

$$\begin{aligned} & \left(\frac{1}{n!} \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(a_{i_j})} \right)^{P_j} \right)^{1/\sum_{j=1}^n P_j} \\ & \leq \left(\frac{1}{n!} \sum_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \prod_{j=1}^n \left(s_{\theta(b_{i_j})} \right)^{P_j} \right)^{1/\sum_{j=1}^n P_j}, \end{aligned} \quad (18)$$

that is, $s_{\theta(a)} \leq s_{\theta(b)}$.

Given that $h_{a_i}^{\sigma(t)} \leq h_{b_i}^{\sigma(t)}$, we can obtain

$$\left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \leq \left(h_{b_{i_j}}^{\sigma(t)} \right)^{P_j}, \quad \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \leq \prod_{j=1}^n \left(h_{b_{i_j}}^{\sigma(t)} \right)^{P_j}. \quad (19)$$

Therefore,

$$\begin{aligned} 1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} & \geq 1 - \prod_{j=1}^n \left(h_{b_{i_j}}^{\sigma(t)} \right)^{P_j}, \quad \prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \\ & \cdot \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \\ & \geq \prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right). \end{aligned} \quad (20)$$

Thus,

$$\begin{aligned} & \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!} \\ & \geq \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!}. \end{aligned} \quad (21)$$

Furthermore,

$$\begin{aligned} & 1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!} \\ & \leq 1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{b_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right)^{1/n!}. \end{aligned} \quad (22)$$

Thus,

$$\begin{aligned} & \left(1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right) \right)^{1/n!} \left(\prod_{j=1}^n P_j \right)^{1/\sum_{j=1}^n P_j} \\ & \leq \left(1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{b_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right) \right)^{1/n!} \left(\prod_{j=1}^n P_j \right)^{1/\sum_{j=1}^n P_j}, \quad (23) \\ & \left(1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{a_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right) \right)^{1/n!} \left(\prod_{j=1}^n P_j \right)^{1/\sum_{j=1}^n P_j} \\ & \leq \left(1 - \left(\prod_{\substack{i_1, i_2, \dots, i_n=1 \\ i_1 \neq i_2 \neq \dots \neq i_n}}^n \left(1 - \prod_{j=1}^n \left(h_{b_{i_j}}^{\sigma(t)} \right)^{P_j} \right) \right) \right)^{1/n!} \left(\prod_{j=1}^n P_j \right)^{1/\sum_{j=1}^n P_j}, \quad (24) \end{aligned}$$

that is, $h(a) \leq h(b)$.

As $s(a) = (1/\#h \sum_{t=1}^h h_{a_i}^{\sigma(t)}) s_{\theta(a)}$ and $s(b) = (1/\#h \sum_{t=1}^h h_{b_i}^{\sigma(t)}) s_{\theta(b)}$, then by (18) and (23), we have $s(a) \leq s(b)$, which completes the proof.

Theorem 3 (idempotency). *If $a_i (i = 1, 2, \dots, n)$ are equal that is, $a_i = a = \langle s_{\theta(a)}, h(a) \rangle$, then*

$$\text{HFLMM}^P(a_1, a_2, \dots, a_n) = a. \quad (25)$$

Proof 3. According to Theorem 1, we can obtain

$$\begin{aligned}
\text{HFLMM}^P(a, a, \dots, a) &= \left\langle \left(\frac{1}{n!} \sum_{\vartheta \in S_n} \prod_{j=1}^n (s_{\vartheta(j)})^{P_j} \right)^{1/\sum_{j=1}^n P_j}, \left\{ \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n (h_{\vartheta(j)}^{\sigma(t)})^{P_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n P_j} \mid t = 1, 2, \dots, h \right\} \right\rangle \\
&= \left\langle \left(\frac{1}{n!} \sum_{\vartheta \in S_n} (s_{\vartheta(a)})^{\sum_{j=1}^n P_j} \right)^{1/\sum_{j=1}^n P_j}, \left\{ \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - (h_a^{\sigma(t)})^{\sum_{j=1}^n P_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n P_j} \mid t = 1, 2, \dots, h \right\} \right\rangle \\
&= \left\langle \left(\frac{1}{n!} \left((n!) \times (s_{\vartheta(a)})^{\sum_{j=1}^n P_j} \right) \right)^{1/\sum_{j=1}^n P_j}, \left\{ \left(1 - \left(\left(1 - (h_a^{\sigma(t)})^{\sum_{j=1}^n P_j} \right)^{n!} \right)^{1/n!} \right)^{1/\sum_{j=1}^n P_j} \mid t = 1, 2, \dots, h \right\} \right\rangle \\
&= \left\langle \left((s_{\vartheta(a)})^{\sum_{j=1}^n P_j} \right)^{1/\sum_{j=1}^n P_j}, \left\{ \left(1 - \left(1 - (h_a^{\sigma(t)})^{\sum_{j=1}^n P_j} \right) \right) \mid t = 1, 2, \dots, h \right\} \right\rangle \\
&= \langle s_{\vartheta(a)}, \{ h_a^{\sigma(t)} \mid t = 1, 2, \dots, h \} \rangle = \langle s_{\vartheta(a)}, h(a) \rangle = a.
\end{aligned} \tag{26}$$

Theorem 4 (boundedness). Let $a_i = \langle s_{\vartheta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs as $\#h(a_i) = h$ holds for all i . If $a^- = \langle \min_i(s_{\vartheta(a_i)}), \{ \min_i(h_{a_i}^{\sigma(t)} \mid t = 1, 2, \dots, h) \} \rangle$ and $a^+ = \langle \max_i(s_{\vartheta(a_i)}), \{ \max_i(h_{a_i}^{\sigma(t)} \mid t = 1, 2, \dots, h) \} \rangle$, then

$$a^- \leq \text{HFLMM}^P(a_1, a_2, \dots, a_n) \leq a^+. \tag{27}$$

Proof 4. According to Theorems 2 and 3, we can obtain

$$\text{HFLMM}^P(a_1, a_2, \dots, a_n) \leq \text{HFLMM}^P(a^+, a^+, \dots, a^+) = a^+, \tag{28}$$

and

$$\text{HFLMM}^P(a_1, a_2, \dots, a_n) \geq \text{HFLMM}^P(a^-, a^-, \dots, a^-) = a^-. \tag{29}$$

Hence, we can obtain $a^- \leq \text{HFLMM}^P(a_1, a_2, \dots, a_n) \leq a^+$.

In the following, we will explore special cases of HFLMM operators with respect to parameter vector P .

Case 1. If $P = (1, 0, 0, \dots, 0)$, then HFLMM is reduced to the hesitant fuzzy linguistic averaging (HFLA) operator.

$$\begin{aligned}
&\text{HFLMM}^P(a_1, a_2, \dots, a_n) \\
&= \frac{1}{n} \left(\bigoplus_{i=1}^n a_i \right) \\
&= \left\langle \frac{1}{n} \sum_{i=1}^n s_{\vartheta(a_i)}, \left\{ 1 - \prod_{i=1}^n \left(1 - h_{a_i}^{\sigma(t)} \right)^{1/n} \mid t = 1, 2, \dots, h \right\} \right\rangle.
\end{aligned} \tag{30}$$

Case 2. If $P = (\lambda, 0, 0, \dots, 0)$, then HFLMM is reduced to the generalized hesitant fuzzy linguistic averaging (GHFLA) operator.

$$\begin{aligned}
&\text{HFLMM}^P(a_1, a_2, \dots, a_n) \\
&= \left(\frac{1}{n} \left(\bigoplus_{i=1}^n (a_i)^\lambda \right) \right)^{1/\lambda} \\
&= \left\langle \left(\frac{1}{n} \sum_{i=1}^n s_{\vartheta(a_i)}^\lambda \right)^{1/\lambda}, \right. \\
&\quad \cdot \left\{ \left(1 - \left(1 - \prod_{i=1}^n \left(1 - (h_{a_i}^{\sigma(t)})^\lambda \right) \right)^{1/n} \right)^{1/\lambda} \right. \\
&\quad \cdot \left. \mid t = 1, 2, \dots, h \right\} \left. \right\rangle.
\end{aligned} \tag{31}$$

Case 3. If $P = (1, 1, 0, 0, \dots, 0)$, then HFLMM is reduced to the hesitant fuzzy linguistic Bonferroni mean (HFLBM) operator.

$$\begin{aligned}
& \text{HFLMM}^P(a_1, a_2, \dots, a_n) \\
&= \left(\frac{1}{n(n-1)} \left(\bigoplus_{\substack{i,j=1 \\ i \neq j}}^n (a_i \otimes a_j) \right) \right)^{1/2} \\
&\cdot \left\langle \left(\frac{1}{n(n-1)} \left(\sum_{\substack{i,j=1 \\ i \neq j}}^n s_{\theta(a_i)} s_{\theta(a_j)} \right) \right)^{1/2} \right. \\
&\cdot \left. \left\{ \left(1 - \prod_{\substack{i,j=1 \\ i \neq j}}^n \left(1 - h_{a_i}^{\sigma(t)} h_{a_j}^{\sigma(t)} \right)^{1/n(n-1)} \right)^{1/2} \right. \right. \\
&\cdot \left. \left. |t = 1, 2, \dots, h \right\} \right\rangle.
\end{aligned} \tag{32}$$

Case 4. If $P = (\overbrace{1, 1, \dots, 1}^k, \overbrace{0, 0, \dots, 0}^{n-k})$, then HFLMM is reduced to the hesitant fuzzy linguistic Maclaurin symmetric mean operator.

$$\begin{aligned}
& \text{HFLMM}^P(a_1, a_2, \dots, a_n) \\
&= \left(\frac{\bigoplus_{1 \leq i_1 < \dots < i_k \leq n} \bigotimes_{j=1}^k (a_{i_j})}{C_n^k} \right)^{1/k} \\
&= \left\langle \left(\frac{1}{C_n^k} \sum_{1 \leq i_1 < \dots < i_k \leq n} \prod_{j=1}^k s_{\theta(a_{i_j})} \right)^{1/k} \right. \\
&\cdot \left. \left\{ \left(1 - \prod_{1 \leq i_1 < \dots < i_k \leq n} \left(1 - \prod_{j=1}^k h_{a_{i_j}}^{\sigma(t)} \right)^{1/C_n^k} \right)^{1/k} \right. \right. \\
&\cdot \left. \left. |t = 1, 2, \dots, h \right\} \right\rangle.
\end{aligned} \tag{33}$$

Case 5. If $P = (1/n, 1/n, \dots, 1/n)$, then HFLMM is reduced to the hesitant fuzzy linguistic geometric averaging (HFLGA) operator.

$$\begin{aligned}
& \text{HFLMM}^P(a_1, a_2, \dots, a_n) \\
&= \bigotimes_{i=1}^n (a_i)^{1/n} = \left\langle \prod_{i=1}^n (s_{\theta(a_i)})^{1/n}, \left\{ \prod_{i=1}^n (h_{a_i}^{\sigma(t)})^{1/n} \mid t = 1, 2, \dots, h \right\} \right\rangle.
\end{aligned} \tag{34}$$

3.2. Hesitant Fuzzy Linguistic Dual Muirhead Mean

Definition 9. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs with $\#h(a_i) = h$ holding for all i , and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then the hesitant fuzzy linguistic dual Muirhead mean (HFLDMM) can be defined as

$$\text{HFLDMM}^P(a_1, a_2, \dots, a_n) = \frac{1}{\sum_{j=1}^n p_j} \left(\bigotimes_{\vartheta \in S_n} \left(\bigoplus_{j=1}^n (p_j a_{\vartheta(j)}) \right) \right)^{1/n!}, \tag{35}$$

where $\vartheta(j) (j = 1, 2, \dots, n)$ is any permutation of $(1, 2, \dots, n)$, and S_n is the collection of all permutations of $(1, 2, \dots, n)$.

Similar to HFLMM operator, we can obtain the following theorem according to Definition 5.

Theorem 5. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs as $\#h(a_i) = h$ holding for all i and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then the aggregated value of HFLDMM is also an HFLE and

$$\begin{aligned}
& \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\
&= \left\langle \frac{1}{\sum_{j=1}^n p_j} \left(\prod_{\vartheta \in S_n} \sum_{j=1}^n p_j s_{\theta_{\vartheta(j)}} \right)^{1/n!} \right. \\
&\cdot \left. \left\{ 1 - \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n p_j} \right. \right. \\
&\cdot \left. \left. |t = 1, 2, \dots, h \right\} \right\rangle.
\end{aligned} \tag{36}$$

Proof 5. According to Definition 5, we have

$$p_j a_i = \left\langle p_j s_{\theta_{\vartheta(j)}}, \left\{ 1 - \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \mid t = 1, 2, \dots, h \right\} \right\rangle, \tag{37}$$

and

$$\bigotimes_{j=1}^n (p_j a_{\vartheta(j)}) = \left\langle \sum_{j=1}^n p_j s_{\theta_{\vartheta(j)}}, \left\{ 1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \mid t = 1, 2, \dots, h \right\} \right\rangle. \tag{38}$$

Therefore,

$$\begin{aligned} & \left(\bigotimes_{\vartheta \in \mathcal{S}_n} \left(\bigoplus_{j=1}^n (p_j a_{\vartheta(j)}) \right) \right) \\ &= \left\langle \prod_{\vartheta \in \mathcal{S}_n} \sum_{j=1}^n p_j s_{\vartheta(j)}, \left\{ \prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right. \right. \\ & \quad \left. \left. \cdot |t = 1, 2, \dots, h \right\} \right\rangle. \end{aligned} \quad (39)$$

Furthermore,

$$\begin{aligned} & \left(\bigotimes_{\vartheta \in \mathcal{S}_n} \left(\bigoplus_{j=1}^n (p_j a_{\vartheta(j)}) \right) \right)^{1/n!} \\ &= \left\langle \left(\prod_{\vartheta \in \mathcal{S}_n} \sum_{j=1}^n p_j s_{\vartheta(j)} \right)^{1/n!}, \right. \\ & \quad \left. \cdot \left\{ \left(\prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right)^{1/n!} \mid t = 1, 2, \dots, h \right\} \right\rangle. \end{aligned} \quad (40)$$

Thus,

$$\begin{aligned} & \frac{1}{\sum_{j=1}^n p_j} \left(\bigotimes_{\vartheta \in \mathcal{S}_n} \left(\bigoplus_{j=1}^n (p_j a_{\vartheta(j)}) \right) \right)^{1/n!} \\ &= \left\langle \frac{1}{\sum_{j=1}^n p_j} \left(\prod_{\vartheta \in \mathcal{S}_n} \sum_{j=1}^n p_j s_{\vartheta(j)} \right)^{1/n!}, \right. \\ & \quad \left. \cdot \left\{ 1 - \left(1 - \left(\prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n p_j} \right. \right. \\ & \quad \left. \left. \cdot |t = 1, 2, \dots, h \right\} \right\rangle. \end{aligned} \quad (41)$$

Hence, (36) is maintained.

Given that $h_{\vartheta(j)}^{\sigma(t)} \in [0, 1]$, then we can obtain

$$\left(1 - h_{\vartheta(j)}^{\sigma(t)} \right) \in [0, 1], \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \in [0, 1], \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \in [0, 1], \quad (42)$$

and then

$$1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \in [0, 1], \prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \in [0, 1]. \quad (43)$$

Therefore,

$$\begin{aligned} & \left(\prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right)^{1/n!} \\ & \in [0, 1], 1 - \left(\prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right)^{1/n!} \in [0, 1]. \end{aligned} \quad (44)$$

Thus,

$$\left(1 - \left(\prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n p_j} \in [0, 1]. \quad (45)$$

Hence,

$$1 - \left(1 - \left(\prod_{\vartheta \in \mathcal{S}_n} \left(1 - \prod_{j=1}^n \left(1 - h_{\vartheta(j)}^{\sigma(t)} \right)^{p_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n p_j} \in [0, 1]. \quad (46)$$

Therefore, (35) is an HFLE, which completes the proof.

Similar to HFLMM, HFLDMM has the following theorems that can be easily proven.

Theorem 6. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs.

(1) *Monotonicity:* let $b_i = \langle s_{\theta(b_i)}, h(b_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs; if these conditions are satisfied for all i : $\#h(a_i) = \#h(b_i) = h$; $s_{\theta(a_i)} \leq s_{\theta(b_i)}$; $h_{a_i}^{\sigma(t)} \leq h_{b_i}^{\sigma(t)}$, where $t = 1, 2, \dots, h$, then

$$\begin{aligned} & \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\ & \leq \text{HFLDMM}^P(b_1, b_2, \dots, b_2). \end{aligned} \quad (47)$$

(2) *Idempotency:* if $a_i (i = 1, 2, \dots, n)$ are equal, that is, $a_i = a = \langle s_{\theta(a)}, h(a) \rangle$, then

$$\begin{aligned} & \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\ & \leq \text{HFLDMM}^P(b_1, b_2, \dots, b_2). \end{aligned} \quad (48)$$

(3) *Boundedness:* if $a^- = \langle \min_i (s_{\theta(a_i)}), \{ \min_i (h_{a_i}^{\sigma(t)} \mid t = 1, 2, \dots, h) \} \rangle$ and $a^+ = \langle \max_i (s_{\theta(a_i)}), \{ \max_i (h_{a_i}^{\sigma(t)} \mid t = 1, 2, \dots, h) \} \rangle$, then

$$a^- \leq \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \leq a^+. \quad (49)$$

In the following section, we will explore special cases of the HFLDMM operator with respect to parameter vector P .

Case 1. If $P = (1, 0, 0, \dots, 0)$, then HFLDMM is reduced to the HFLGA operator.

$$\begin{aligned} & \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\ &= \frac{n}{\otimes_{i=1}^n (a_i)^{1/n}} = \left\langle \prod_{i=1}^n (s_{\theta(a_i)})^{1/n}, \left\{ \prod_{i=1}^n (h_{a_i}^{\sigma(t)})^{1/n} \mid t = 1, 2, \dots, h \right\} \right\rangle. \end{aligned} \quad (50)$$

Case 2. If $P = (\lambda, 0, 0, \dots, 0)$, then HFLDMM is reduced to the generalized hesitant fuzzy linguistic geometric averaging (GHFLGA) operator.

$$\begin{aligned} & \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\ &= \frac{1}{\lambda} \left(\otimes_{i=1}^n (\lambda a_i)^{1/n} \right) \\ &= \left\langle \frac{1}{\lambda} \prod_{i=1}^n (\lambda s_{\theta(a_i)})^{1/n}, \right. \\ & \quad \left. \cdot \left\{ 1 - \left(1 - \prod_{i=1}^n \left(1 - \left(1 - h_{a_i}^{\sigma(t)} \right)^\lambda \right)^{1/n} \right)^{1/\lambda} \right\} \right\rangle. \end{aligned} \quad (51)$$

Case 3. If $P = (1, 1, 0, 0, \dots, 0)$, then HFLDMM is reduced to the hesitant fuzzy linguistic geometric Bonferroni mean (HFLGBM) operator.

$$\begin{aligned} & \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\ &= \frac{1}{2} \left(\begin{array}{c} n \\ i, j=1 \\ i \neq j \end{array} (a_i \oplus a_j) \right)^{1/n(n-1)} \\ & \quad \cdot \left\langle \left(\prod_{\substack{i, j=1 \\ i \neq j}}^n (s_{\theta(a_i)} + s_{\theta(a_j)}) \right)^{1/n(n-1)}, \right. \\ & \quad \left. \cdot 1 - \left(1 - \prod_{\substack{i, j=1 \\ i \neq j}}^n (h_i^{\sigma(t)} + h_j^{\sigma(t)} - h_i^{\sigma(t)} h_j^{\sigma(t)})^{1/n(n-1)} \right)^2 \right\rangle. \end{aligned} \quad (52)$$

Case 4. If $P = (\overbrace{1, 1, \dots, 1}^k, \overbrace{0, 0, \dots, 0}^{n-k})$, then HFLDMM is reduced to the hesitant fuzzy linguistic geometric Maclaurin symmetric mean (HFLGMSM) operator.

$$\begin{aligned} & \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\ &= \frac{1}{k} \left(\otimes_{1 \leq i_1 \leq \dots \leq i_k \leq n} \left(\oplus_{j=1}^k a_{i_j} \right)^{1/C_n^k} \right) = \left\langle \frac{1}{k} \prod_{1 \leq i_1 \leq \dots \leq i_k \leq n} \left(\sum_{j=1}^k s_{\theta(a_{i_j})} \right)^{1/C_n^k}, \right. \\ & \quad \left. \cdot 1 - \left(1 - \prod_{1 \leq i_1 \leq \dots \leq i_k \leq n} \left(1 - \prod_{j=1}^k (1 - h_{a_{i_j}}^{\sigma(t)}) \right)^{1/C_n^k} \right)^{1/k} \right\rangle. \end{aligned} \quad (53)$$

Case 5. If $P = (1/n, 1/n, \dots, 1/n)$, then HFLDMM is reduced to the HFLA operator.

$$\begin{aligned} & \text{HFLDMM}^P(a_1, a_2, \dots, a_n) \\ &= \frac{1}{n} \left(\oplus_{i=1}^n a_i \right) = \left\langle \frac{1}{n} \sum_{i=1}^n s_{\theta(a_i)}, \left\{ 1 - \prod_{i=1}^n (1 - h_{a_i}^{\sigma(t)})^{1/n} \mid t = 1, 2, \dots, h \right\} \right\rangle. \end{aligned} \quad (54)$$

4. Hesitant Fuzzy Linguistic Weighted Muirhead Mean Operators

Evidently, HFLMM and the HFLDMM do not consider the weights of the associated HFLEs. Therefore, we develop hesitant fuzzy linguistic weighted Muirhead mean operators that consider the weights of HFLEs.

Definition 10. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs with $\#h(a_i) = h$ holding for all i , and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters. The weight vector is $w = (w_1, w_2, \dots, w_n)^T$, satisfying $w_i \in [0, 1] (i = 1, 2, \dots, n)$ and $\sum_{i=1}^n w_i = 1$. If

$$\begin{aligned} & \text{HFLWMM}^P(a_1, a_2, \dots, a_n) \\ &= \left(\frac{1}{n!} \sum_{\vartheta \in S_n} \left(\otimes_{j=1}^n (n w_{\vartheta(j)} a_{\vartheta(j)})^{p_j} \right) \right)^{1/\sum_{j=1}^n p_j}, \end{aligned} \quad (55)$$

then HFLWMM^P is the hesitant fuzzy linguistic weighted Muirhead mean (HFLWMM), where $\vartheta(j) (j = 1, 2, \dots, n)$ is any permutation of $(1, 2, \dots, n)$ and S_n is the collection of all permutations of $(1, 2, \dots, n)$.

According to Definition 5, we can obtain the following theorem.

Theorem 7. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs with $\#h(a_i) = h$ holding for all i and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then

$$\begin{aligned} & \text{HFLWMM}^P(a_1, a_2, \dots, a_n) \\ &= \left\langle \left(\frac{1}{n!} \sum_{\vartheta \in S_n} \prod_{j=1}^n (n w_{\vartheta(j)} s_{\theta(a_{\vartheta(j)})})^{p_j} \right)^{1/\sum_{j=1}^n p_j}, \right. \\ & \quad \cdot \left\{ \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n (1 - (1 - h_{\vartheta(j)}^{\sigma(t)})^{n w_{\vartheta(j)}})^{p_j} \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n p_j} \right. \\ & \quad \left. \cdot |t = 1, 2, \dots, h \right\} \right\rangle. \end{aligned} \quad (56)$$

The proof of Theorem 7 is similar to that of Theorem 1 and is thus omitted to save space.

Definition 11. Let $a_i = \langle s_{\theta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs with $\#h(a_i) = h$ holds for all i and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters. The weight

vector $\text{bew} = (w_1, w_2, \dots, w_n)^T$, satisfying $w_i \in [0, 1] (i = 1, 2, \dots, n)$ and $\sum_{i=1}^n w_i = 1$. If

$$\text{HFLWDMMP}^P(a_1, a_2, \dots, a_n) = \frac{1}{\sum_{j=1}^n P_j} \left(\prod_{\vartheta \in S_n} \sum_{j=1}^n (p_j a_{\vartheta(j)}^{nw_{\vartheta(j)}}) \right)^{1/n!}, \quad (57)$$

then we call HFLWDMMP the hesitant fuzzy linguistic weighted dual Muirhead mean (HFLWDMM) operator, where $\vartheta(j) (j = 1, 2, \dots, n)$ is any permutation of $(1, 2, \dots, n)$, and S_n is the collection of all permutations of $(1, 2, \dots, n)$.

According to Definition 5, we can obtain the following theorem.

Theorem 8. Let $a_i = \langle s_{\vartheta(a_i)}, h(a_i) \rangle (i = 1, 2, \dots, n)$ be a collection of HFLEs with $\#h(a_i) = h$ holds for all i and $P = (p_1, p_2, \dots, p_n) \in R^n$ be a vector of parameters, then the aggregated value of HFLWDMM is also an HFLE and

$$\begin{aligned} & \text{HFLWDMMP}^P(a_1, a_2, \dots, a_n) \\ &= \left\langle \frac{1}{\sum_{j=1}^n P_j} \left(\prod_{\vartheta \in S_n} \sum_{j=1}^n P_j (s_{\vartheta(j)})^{nw_{\vartheta(j)}} \right)^{1/n!}, \right. \\ & \cdot \left. \left\{ 1 - \left(1 - \left(\prod_{\vartheta \in S_n} \left(1 - \prod_{j=1}^n \left(1 - (h_{\vartheta(j)}^{\sigma(t)})^{nw_{\vartheta(j)}} P_j \right) \right) \right)^{1/n!} \right)^{1/\sum_{j=1}^n P_j} \right. \right. \\ & \cdot \left. \left. |t = 1, 2, \dots, h \right\} \right\rangle. \end{aligned} \quad (58)$$

The proof of Theorem 7 is similar to that of Theorem 1 and is thus omitted.

5. Novel Approach to MAGDM with Hesitant Fuzzy Linguistic Information

In this section, we propose a novel approach to MAGDM based on the proposed aggregation operators. A typical MAGDM problem, wherein the attribute values take the form of HFLEs, can be described as follows: let $X = \{x_1, x_2, \dots, x_m\}$ be a set of alternatives and $G = \{G_1, G_2, \dots, G_n\}$ be n attributes with the weight vector being $w = (w_1, w_2, \dots, w_n)^T$, satisfying $w_i \in [0, 1]$ and $\sum_{i=1}^n w_i = 1$. A set of experts is organized to act as decision-makers and for attribute $G_j (j = 1, 2, \dots, n)$ of alternative $x_i (i = 1, 2, \dots, m)$. Decision-makers are required to express their assessments anonymously by using an HFLE that can be denoted by $a_{ij} = \langle s_{\vartheta(a_{ij})}, h(a_{ij}) \rangle$. Therefore, the hesitant fuzzy linguistic decision matrix expressed as $A = (a_{ij})_{m \times n}$ can be obtained. We propose a new method to MAGDM in the following.

Step 1. Normalize the decision matrix. The original decision matrix should be normalized from two points of view. First,

if attributes can be divided into benefit and cost types, then the decision matrix should be normalized by

$$a_{ij} = \langle s_{\vartheta(a_{ij})}, h(a_{ij}) \rangle = \begin{cases} \langle s_{\vartheta \in (a_{ij})}, h(a_{ij}) \rangle & G_j \in I_1 \\ \langle s_{\vartheta \in (a_{ij})}, 1 - h(a_{ij}) \rangle & G_j \in I_2 \end{cases}, \quad (59)$$

where I_1 and I_2 denote the benefit and cost types, respectively. Second, we assume that all the HFEs in HFLEs have the same number of values to simplify the calculation process. Thus, we can extend the short HFEs of the corresponding HFLEs according to the transformation regulation presented in Definition 5 until all HFEs have the same number of values.

Step 2. For alternative $x_i (i = 1, 2, \dots, m)$, utilize the HFLWMM operator

$$a_i = \text{HFLWMM}^P(a_{i1}, a_{i2}, \dots, a_{in}), \quad (60)$$

or the HFLDWMM operator

$$a_i = \text{HFLWDMMP}^P(a_{i1}, a_{i2}, \dots, a_{in}), \quad (61)$$

to aggregate all the preference information provided by decision-makers.

Step 3. Calculate the scores of the overall values $a_i (i = 1, 2, \dots, m)$ and rank them according to Definition 3.

Step 4. Rank the alternatives $x_i (i = 1, 2, \dots, m)$ according to the corresponding rank of overall values and select the best alternative.

6. Numerical Example

In this section, we provide a numerical example adopted from [31] to validate the proposed approach. A company decides to implement an enterprise resource planning (ERP) system. After primary evaluation, five potential ERP systems $x_i (i = 1, 2, 3, 4, 5)$ are chosen as candidates. To select the best ERP system, the company invites several professional experts to aid in this decision-making. The candidates are from four aspects (attributes), namely, (1) functionality and technology G_1 , (2) strategic fitness G_2 , (3) vendor's ability G_3 , and (4) vendor's reputation G_4 . The weight vector of the attributes is $w = (0.2, 0.15, 0.35, 0.3)^T$. Decision-makers are required to evaluate the five possible ERP systems anonymously. A hesitant fuzzy linguistic decision matrix $D = (a_{ij})_{5 \times 4} = ((s_{\vartheta(a_{ij})}, h(a_{ij})))_{5 \times 4}$ is presented in Table 1, where $\langle s_{\vartheta(a_{ij})}, h(a_{ij}) \rangle (i = 1, 2, 3, 4, 5; j = 1, 2, 3, 4)$ is a series of HFLEs. In the following section, we utilize the proposed approach to solve the problem.

6.1. Decision-Making Process

Step 1. Normalize the decision matrix. All the attributes are benefit attributes and thus, we only need to extend the short

TABLE 1: Original hesitant fuzzy linguistic decision matrix.

	G_1	G_2	G_3	G_4
x_1	$\langle s_4, \{0.4, 0.5, 0.6\} \rangle$	$\langle s_2, \{0.6, 0.7\} \rangle$	$\langle s_3, \{0.3, 0.4, 0.7, 0.9\} \rangle$	$\langle s_7, \{0.2, 0.4\} \rangle$
x_2	$\langle s_2, \{0.2, 0.3, 0.4, 0.6\} \rangle$	$\langle s_7, \{0.5, 0.7\} \rangle$	$\langle s_2, \{0.4, 0.5, 0.7\} \rangle$	$\langle s_5, \{0.7, 0.8\} \rangle$
x_3	$\langle s_5, \{0.5, 0.6, 0.7, 0.8\} \rangle$	$\langle s_4, \{0.2, 0.3, 0.5\} \rangle$	$\langle s_6, \{0.6, 0.8\} \rangle$	$\langle s_6, \{0.4, 0.6\} \rangle$
x_4	$\langle s_6, \{0.5, 0.6, 0.7\} \rangle$	$\langle s_3, \{0.2, 0.5\} \rangle$	$\langle s_1, \{0.8, 0.9\} \rangle$	$\langle s_2, \{0.3, 0.4, 0.5, 0.8\} \rangle$
x_5	$\langle s_1, \{0.3, 0.6, 0.7\} \rangle$	$\langle s_5, \{0.3, 0.4\} \rangle$	$\langle s_4, \{0.7, 0.8\} \rangle$	$\langle s_3, \{0.2, 0.3, 0.4\} \rangle$

TABLE 2: Normalized hesitant fuzzy linguistic decision matrix.

	G_1	G_2	G_3	G_4
x_1	$\langle s_4, \{0.4, 0.5, 0.6, 0.6\} \rangle$	$\langle s_2, \{0.6, 0.7, 0.7, 0.7\} \rangle$	$\langle s_3, \{0.3, 0.4, 0.7, 0.9\} \rangle$	$\langle s_7, \{0.2, 0.4, 0.4, 0.4\} \rangle$
x_2	$\langle s_2, \{0.2, 0.3, 0.4, 0.6\} \rangle$	$\langle s_7, \{0.5, 0.7, 0.7, 0.7\} \rangle$	$\langle s_2, \{0.4, 0.5, 0.7, 0.7\} \rangle$	$\langle s_5, \{0.7, 0.8, 0.8, 0.8\} \rangle$
x_3	$\langle s_5, \{0.5, 0.6, 0.7, 0.8\} \rangle$	$\langle s_4, \{0.2, 0.3, 0.5, 0.5\} \rangle$	$\langle s_6, \{0.6, 0.8, 0.8, 0.8\} \rangle$	$\langle s_6, \{0.4, 0.6, 0.6, 0.6\} \rangle$
x_4	$\langle s_6, \{0.5, 0.6, 0.7, 0.7\} \rangle$	$\langle s_3, \{0.2, 0.5, 0.5, 0.5\} \rangle$	$\langle s_1, \{0.8, 0.9, 0.9, 0.9\} \rangle$	$\langle s_2, \{0.3, 0.4, 0.5, 0.8\} \rangle$
x_5	$\langle s_1, \{0.3, 0.6, 0.7, 0.7\} \rangle$	$\langle s_5, \{0.3, 0.4, 0.4, 0.4\} \rangle$	$\langle s_4, \{0.7, 0.8, 0.8, 0.8\} \rangle$	$\langle s_3, \{0.2, 0.3, 0.4, 0.4\} \rangle$

HFLs by adding certain values. Here, we assume that decision-makers are optimistic. Therefore, we can obtain the normalized hesitant fuzzy linguistic decision matrix, as shown in Table 2.

Step 2. For each alternative $x_i (i = 1, 2, 3, 4, 5)$, (60) is used to aggregate all the attribute values. Therefore, we can obtain a set of overall values. Without loss of generality, we assume $P = (1, 1, 1, 1)$.

$$\begin{aligned}
a_1 &= \langle s_{3.4117}, 0.3383, 0.4758, 0.5620, 0.5856 \rangle, \\
a_2 &= \langle s_{3.2596}, 0.3839, 0.5103, 0.5888, 0.6569 \rangle, \\
a_3 &= \langle s_{4.9087}, 0.3646, 0.4944, 0.5953, 0.6193 \rangle, \\
a_4 &= \langle s_{2.3212}, 0.3591, 0.5281, 0.5812, 0.6447 \rangle, \\
a_5 &= \langle s_{2.6374}, 0.3093, 0.4547, 0.5087, 0.5087 \rangle.
\end{aligned} \tag{62}$$

Step 3. Calculate the scores of $a_i (i = 1, 2, 3, 4, 5)$ and rank them. According to Definition 3, we can derive $s(a_1) = 1.6732$ $s(a_2) = 1.7438$ $s(a_3) = 2.5446$ $s(a_4) = 1.2262$ $s(a_5) = 1.1745$. Therefore, the rank order of the overall values is $a_3 > a_2 > a_1 > a_4 > a_5$.

Step 4. Rank alternatives $x_i (i = 1, 2, 3, 4, 5)$ according to the rank of $a_i (i = 1, 2, 3, 4, 5)$, that is, $x_3 > x_2 > x_1 > x_4 > x_5$. Therefore, x_3 is the best ERP system. In Step 2, if we utilize (61) to aggregate attribute values, then we can obtain a series of overall values

$$\begin{aligned}
a_1 &= \langle s_{4.8833}, 0.4440, 0.5541, 0.6932, 0.7227 \rangle, \\
a_2 &= \langle s_{3.6232}, 0.5010, 0.6369, 0.6899, 0.7221 \rangle, \\
a_3 &= \langle s_{6.6983}, 0.4526, 0.6183, 0.6808, 0.7124 \rangle, \\
a_4 &= \langle s_{2.3559}, 0.5178, 0.6803, 0.7156, 0.7716 \rangle, \\
a_5 &= \langle s_{3.5820}, 0.4282, 0.5870, 0.6298, 0.6298 \rangle.
\end{aligned} \tag{63}$$

TABLE 3: Results by using different aggregation operators.

Aggregation operators	Values of parameters	Ranking results
HFLWA [31]	No parameters	$x_3 > x_2 > x_1 > x_4 > x_5$
HFLWG [31]	No parameters	$x_3 > x_2 > x_1 > x_5 > x_4$
HFLWHA [40, 41]	No parameters	$x_3 > x_1 > x_2 > x_5 > x_4$
HFLPWA [43]	No parameters	$x_3 > x_1 > x_2 > x_5 > x_4$
HFLPWG [43]	No parameters	$x_3 > x_1 > x_2 > x_5 > x_4$
HFLWBM [42]	$p = 2, q = 3$	$x_3 > x_2 > x_1 > x_4 > x_5$
HFLWGMBM [42]	$p = 2, q = 3$	$x_3 > x_1 > x_2 > x_4 > x_5$

The scores of the overall values are as follows:

$$\begin{aligned}
s(a_1) &= 2.8812 \quad s(a_2) = 2.3097 \quad s(a_3) = 4.1263 \quad s(a_4) \\
&= 1.5816 \quad s(a_5) = 2.0371.
\end{aligned} \tag{64}$$

Therefore, we can obtain $x_3 > x_1 > x_2 > x_5 > x_4$, where x_3 is the best alternative.

6.2. Further Discussion. To demonstrate the effectiveness of the proposed approach, we utilize other methods to solve the above example. These methods include the ones proposed by Lin et al. [31] based on hesitant fuzzy linguistic weighted averaging (HFLW) operator and HFLW geometric (HFLWG) operator, by Lin et al. [40] and Wei et al. [41] based on hesitant fuzzy linguistic hybrid average operator, by Lin et al. [31] based on hesitant fuzzy linguistic prioritized weighted average (HFLPWA) operator and hesitant fuzzy linguistic prioritized weighted geometric (HFLPWG) operator, and by Liu [42] based on hesitant fuzzy linguistic weighted Bonferroni mean (HFLWBM) operator and hesitant fuzzy linguistic weighted geometric Bonferroni mean (HFLWGBM) operator. Details are presented in Table 3.

TABLE 4: Comparison of different approaches and aggregation operators.

Approaches	Whether the interrelationship of two attributes is captured	Whether the interrelationship of three attributes is captured	Whether the interrelationship of multiple attributes is captured	Whether it makes the method flexible by the parameter vector
HFLWA [31]	No	No	No	No
HFLWG [31]	No	No	No	No
HFLWHA [40, 41]	No	No	No	No
HFLPWA [43]	Yes	Yes	Yes	No
HFPLWG [43]	Yes	Yes	Yes	No
HFLWBM [42]	Yes	No	No	No
HFLWGBM [42]	Yes	No	No	No
HFLWMM	Yes	Yes	Yes	Yes
HFLWDMM	Yes	Yes	Yes	Yes

As seen from Table 3, if we use the HFLWA and the HFLWBM operators, we can obtain the same ranking result as that derived by the HFLWMM operator. If we use the HFLWHA, HFLPWA, and HFPLWG operators, we can obtain the same ranking result as that obtained by the HFLWDMM operator. Therefore, the proposed approach is effective in handling MAGDM with hesitant fuzzy linguistic information. To further demonstrate the merits and superiorities of the newly developed approach, we compare our method with those in [31, 41–43]. Table 4 presents several characteristics of these operators.

Through a comparison with other approaches and aggregation operators, we draw the following conclusions. The weaknesses of the approaches based on HFLWA and HFLWG are that (1) the calculation is based on the operational laws in Definition 4, and they are too complicated to use and (2) the methods cannot capture the interrelationship among arguments. The proposed approach in this paper is more general and flexible than those based on HFLWA and HFLWG. First, the new decision-making approach is based on the new operational laws in Definition 5, resulting in simple calculation. Second, the new method considers the interrelationship among all the arguments. Third, the new approach is more flexible than those based on the HFLWA and HFLWG operators, as HFLWA is a special case of HFLWMM, and HFLWG is a special case of HFLWDMM.

The HFLPWA and HFPLWG operators can consider the entire interrelationship among the HFLEs being fused but can only be used to address the problems with unknown attribute weights. Moreover, the HFLPWA and HFPLWG operators are not as flexible as the HFLWMM and HFLWDMM operators, which have parameter weight P .

The HFLWBM and HFLWGBM operators can only consider the interrelationship between any two arguments. However, in real decision-making problems, all the arguments provided by decision-makers are correlated, which means that interrelationships among all the arguments should be considered. The HFLWMM and HFLWDMM operators can capture all the correlations and consider parameter weight, resulting in flexible information aggregation.

In summary, the proposed HFLWMM and HFLWDMM exhibit the following advantages: (1) these operators are

based on simplified operational laws that streamline the decision-making process, (2) they consider the interrelationship among all arguments, and (3) they have parameter vector P , which leads to flexible information aggregation.

Parameter vector P plays an important role in the ranking results. To demonstrate this, we investigate different cases by assigning different values to P . Additional details can be found in Tables 5 and 6.

As shown in Tables 5 and 6, different ranking results can be obtained by assigning different values to parameter vector P . The HFLWMM and HFLWDMM operators are two flexible aggregation operators. As shown in Table 5, the more interrelationships between attribute values are considered in the HFLWMM operator, the smaller the scores of the overall values are. As seen in Table 6, the more interrelationships between attribute values are considered in the HFLWDMM operator, the higher the scores of the overall values are. Another interesting property is observed when all the values in parameter P are equal, that is, the scores of overall values and the ranking results are the same regardless of the values.

7. Conclusions

The HFLS is a powerful and efficient tool for describing the fuzziness, uncertainty, and hesitancy of decision-makers in MAGDM. In this paper, we introduced a new approach to MAGDM with hesitant fuzzy linguistic information. First, we investigated the MM under hesitant fuzzy linguistic environment and introduced the HFLMM, HFLDMM, HFLWMM, and HFLWDMM operators. These operators considered the interrelationship among all HFLEs. Moreover, the presence of the parameter vector leads to flexible information aggregation process. Second, we proposed a novel approach to MAGDM by using these operators. Third, we provided a numerical example and performed comparative analysis to illustrate the validity and advantages of the new approach. The proposed approach can be used to effectively solve MAGDM problems with hesitant fuzzy linguistic information. In the future, we will apply the proposed method to real decision problems. In addition, we will

TABLE 5: Ranking results by assigning different values to parameter vector P in the HFLWMM operator.

Parameter P	Scores of overall values	Ranking results
$P = (1, 0, 0, 0)$	$s(a_1) = 2.3070s(a_2) = 2.3131s(a_3) = 3.5003s(a_4) = 1.9661s(a_5) = 1.9116$	$x_3 \succ x_2 \succ x_1 \succ x_4 \succ x_5$
$P = (1, 1, 0, 0)$	$s(a_1) = 1.9932s(a_2) = 2.0440s(a_3) = 3.0597s(a_4) = 1.4525s(a_5) = 1.5181$	$x_3 \succ x_2 \succ x_1 \succ x_5 \succ x_4$
$P = (1, 1, 1, 0)$	$s(a_1) = 1.8304s(a_2) = 1.8805s(a_3) = 2.7951s(a_4) = 1.3215s(a_5) = 1.3489$	$x_3 \succ x_2 \succ x_1 \succ x_5 \succ x_4$
$P = (1, 1, 1, 1)$	$s(a_1) = 1.6732s(a_2) = 1.7438s(a_3) = 2.5446s(a_4) = 1.2262s(a_5) = 1.1745$	$x_3 \succ x_2 \succ x_1 \succ x_4 \succ x_5$
$P = (0.5, 0.5, 0.5, 0.5)$	$s(a_1) = 1.6732s(a_2) = 1.7438s(a_3) = 2.5446s(a_4) = 1.2262s(a_5) = 1.1745$	$x_3 \succ x_2 \succ x_1 \succ x_4 \succ x_5$
$P = (2, 2, 2, 2)$	$s(a_1) = 1.6732s(a_2) = 1.7438s(a_3) = 2.5446s(a_4) = 1.2262s(a_5) = 1.1745$	$x_3 \succ x_2 \succ x_1 \succ x_4 \succ x_5$
$P = (3, 3, 3, 3)$	$s(a_1) = 1.6732s(a_2) = 1.7438s(a_3) = 2.5446s(a_4) = 1.2262s(a_5) = 1.1745$	$x_3 \succ x_2 \succ x_1 \succ x_4 \succ x_5$

TABLE 6: Ranking results by assigning different values to parameter vector P in the HFLWDMM operator.

Parameter P	Scores of overall values	Ranking results
$P = (1, 0, 0, 0)$	$s(a_1) = 1.8965s(a_2) = 1.8511s(a_3) = 3.2432s(a_4) = 1.2723s(a_5) = 1.4413$	$x_3 \succ x_1 \succ x_2 \succ x_5 \succ x_4$
$P = (1, 1, 0, 0)$	$s(a_1) = 2.4262s(a_2) = 2.0996s(a_3) = 3.7868s(a_4) = 1.4526s(a_5) = 1.8207$	$x_3 \succ x_1 \succ x_2 \succ x_5 \succ x_4$
$P = (1, 1, 1, 0)$	$s(a_1) = 2.6944s(a_2) = 2.2282s(a_3) = 4.0105$ $s(a_4) = 1.5287s(a_5) = 1.9569$	$x_3 \succ x_1 \succ x_2 \succ x_5 \succ x_4$
$P = (1, 1, 1, 1)$	$s(a_1) = 2.8812s(a_2) = 2.3097s(a_3) = 4.1263$ $s(a_4) = 1.5816s(a_5) = 2.0371$	$x_3 \succ x_1 \succ x_2 \succ x_5 \succ x_4$
$P = (0.5, 0.5, 0.5, 0.5)$	$s(a_1) = 2.8812s(a_2) = 2.3097s(a_3) = 4.1263$ $s(a_4) = 1.5816s(a_5) = 2.0371$	$x_3 \succ x_1 \succ x_2 \succ x_5 \succ x_4$
$P = (2, 2, 2, 2)$	$s(a_1) = 2.8812s(a_2) = 2.3097s(a_3) = 4.1263$ $s(a_4) = 1.5816s(a_5) = 2.0371$	$x_3 \succ x_1 \succ x_2 \succ x_5 \succ x_4$
$P = (3, 3, 3, 3)$	$s(a_1) = 2.8812s(a_2) = 2.3097s(a_3) = 4.1263$ $s(a_4) = 1.5816s(a_5) = 2.0371$	$x_3 \succ x_1 \succ x_2 \succ x_5 \succ x_4$

investigate more aggregation operators for fusing hesitant fuzzy linguistic information.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by a key program of the National Natural Science Foundation of China (NSFC) with grant number 71532002 and the Fundamental Research Funds for the Central Universities with grant number 2017YJS075.

References

[1] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
 [2] J. M. Merigó, A. M. Gil-Lafuente, and R. R. Yager, “An overview of fuzzy research with bibliometric indicators,” *Applied Soft Computing*, vol. 27, pp. 420–433, 2015.
 [3] I. B. Turksen, “Interval valued fuzzy sets based on normal forms,” *Fuzzy Sets and Systems*, vol. 20, no. 2, pp. 191–210, 1986.

[4] K. T. Atanassov, “Intuitionistic fuzzy sets,” *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.
 [5] K. Atanassov and G. Gargov, “Interval valued intuitionistic fuzzy sets,” *Fuzzy Sets and Systems*, vol. 31, no. 3, pp. 343–349, 1989.
 [6] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, NY, USA, 1980.
 [7] F. Smarandache, *A Unifying Field in Logics. Neutrosophy: Neutrosophic Probability, Set and Logic*, American Research Press, Rehoboth, DE, USA, 1998.
 [8] S. M. Chen and C. H. Chang, “Fuzzy multi-attribute decision making based on transformation techniques of intuitionistic fuzzy values and intuitionistic fuzzy geometric averaging operators,” *Information Sciences*, vol. 352–353, pp. 133–149, 2016.
 [9] P. Liu, J. Liu, and S.-M. Chen, “Some intuitionistic fuzzy Dombi Bonferroni mean operators and their application to multi-attribute group decision making,” *Journal of the Operational Research Society*, vol. 69, no. 1, pp. 1–24, 2018.
 [10] J. Lin, F. Meng, R. Chen, and Q. Zhang, “Preference attitude-based method for ranking intuitionistic fuzzy numbers and its application in renewable energy selection,” *Complexity*, vol. 2018, Article ID 6251384, 14 pages, 2018.
 [11] V. Lakshmana Gomathi Nayagam, S. Jeevaraj, and S. Geetha, “Total ordering for intuitionistic fuzzy numbers,” *Complexity*, vol. 21, no. S2, 66 pages, 2016.
 [12] P. Liu and F. Teng, “Multiple criteria decision making method based on normal interval-valued intuitionistic fuzzy generalized aggregation operator,” *Complexity*, vol. 21, no. 5, 290 pages, 2016.

- [13] P. Liu and S.-M. Chen, "Group decision making based on Heronian aggregation operators of intuitionistic fuzzy numbers," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2514–2530, 2017.
- [14] S. Wan, F. Wang, and J. Dong, "A three-phase method for group decision making with interval-valued intuitionistic fuzzy preference relations," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 998–1010, 2018.
- [15] D. Liu, X. Chen, and D. Peng, "Interval-valued intuitionistic fuzzy ordered weighted cosine similarity measure and its application in investment decision-making," *Complexity*, vol. 2017, Article ID 1891923, 11 pages, 2017.
- [16] Z. H. Yi and H. Q. Li, "Triangular norm-based cuts and possibility characteristics of triangular intuitionistic fuzzy numbers for decision making," *International Journal of Intelligent Systems*, vol. 33, no. 6, pp. 1165–1179, 2018.
- [17] P. Liu and P. Wang, "Some improved linguistic intuitionistic fuzzy aggregation operators and their applications to multiple-attribute decision making," *International Journal of Information Technology & Decision Making*, vol. 16, no. 03, pp. 817–850, 2017.
- [18] P. Liu and X. Liu, "Multiattribute group decision making methods based on linguistic intuitionistic fuzzy power Bonferroni mean operators," *Complexity*, vol. 2017, Article ID 3571459, 15 pages, 2017.
- [19] T. Wu, X. Liu, and F. Liu, "An interval type-2 fuzzy TOPSIS model for large scale group decision making problems with social network information," *Information Sciences*, vol. 432, pp. 392–410, 2018.
- [20] Z. Zhang, "Trapezoidal interval type-2 fuzzy aggregation operators and their application to multiple attribute group decision making," *Neural Computing and Applications*, vol. 29, no. 4, pp. 1039–1054, 2018.
- [21] S. M. Chen and C. Y. Wang, "Fuzzy decision making systems based on interval type-2 fuzzy sets," *Information Sciences*, vol. 242, pp. 1–21, 2013.
- [22] J. Paul and S. J. John, "Type 2 fuzzy multisets and its applications in decision making problems," *Journal of Intelligent & Fuzzy Systems*, vol. 30, no. 1, pp. 359–369, 2015.
- [23] P. Liu and L. Shi, "Some neutrosophic uncertain linguistic number Heronian mean operators and their application to multi-attribute group decision making," *Neural Computing and Applications*, vol. 28, no. 5, pp. 1079–1093, 2017.
- [24] P. Liu and S.-M. Chen, "Multiattribute group decision making based on intuitionistic 2-tuple linguistic information," *Information Sciences*, vol. 430–431, pp. 599–619, 2018.
- [25] X. Peng and J. Dai, "Approaches to single-valued neutrosophic MADM based on MABAC, TOPSIS and new similarity measure with score function," *Neural Computing and Applications*, vol. 29, no. 10, pp. 939–954, 2018.
- [26] P. Liu, T. Mahmood, and Q. Khan, "Group decision making based on power Heronian aggregation operators under linguistic neutrosophic environment," *International Journal of Fuzzy Systems*, vol. 20, no. 3, pp. 970–985, 2018.
- [27] V. Torra, "Hesitant fuzzy sets," *International Journal of Intelligent Systems*, vol. 25, no. 6, pp. 529–539, 2010.
- [28] R. M. Rodríguez, L. Martínez, and F. Herrera, "Hesitant fuzzy linguistic term sets for decision making," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 109–119, 2012.
- [29] J.-t. Wu, J.-q. Wang, J. Wang, H.-y. Zhang, and X.-h. Chen, "Hesitant fuzzy linguistic multicriteria decision-making method based on generalized prioritized aggregation operator," *The Scientific World Journal*, vol. 2014, Article ID 645341, 16 pages, 2014.
- [30] J. Q. Wang and J. J. Li, "The multi-criteria group decision making method based on multi-granularity intuitionistic two semantics," *Science & Technology Information*, vol. 33, no. 1, pp. 8–9, 2009.
- [31] R. Lin, X. F. Zhang, H. J. Wang, and G. W. Wei, "Hesitant fuzzy linguistic aggregation operators and their application to multiple attribute decision making," *Journal of Intelligent & Fuzzy Systems*, vol. 27, no. 1, pp. 49–63, 2014.
- [32] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decision making," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.
- [33] Z. Xu, "Intuitionistic fuzzy aggregation operators," *IEEE Transactions on Fuzzy Systems*, vol. 15, no. 6, pp. 1179–1187, 2007.
- [34] M. Xia and Z. Xu, "Hesitant fuzzy information aggregation in decision making," *International Journal of Approximate Reasoning*, vol. 52, no. 3, pp. 395–407, 2011.
- [35] D. Yu, W. Zhang, and G. Huang, "Dual hesitant fuzzy aggregation operators," *Technological and Economic Development of Economy*, vol. 22, no. 2, pp. 194–209, 2016.
- [36] R. R. Yager and D. P. Filev, "Induced ordered weighted averaging operators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 2, pp. 141–450, 1999.
- [37] R. R. Yager, "Generalized OWA aggregation operators," *Fuzzy Optimization and Decision Making*, vol. 3, no. 1, pp. 93–107, 2004.
- [38] C. Bonferroni, "Sulle medie multiple di potenze," *Bollettino dell'Unione Matematica Italiana*, vol. 5, no. 3-4, pp. 267–270, 1950.
- [39] S. Sykora, *Mathematical Means and Averages: Generalized Heronian Means*, Library, S. Sykora, Stans, Switzerland, 2009.
- [40] R. Lin, X. F. Zhao, and G. W. Wei, "Models for selecting an ERP system with hesitant fuzzy linguistic information," *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 5, pp. 2155–2165, 2014.
- [41] G. W. Wei, H. T. Alsaadi, and A. Alsaedi, "Hesitant fuzzy linguistic arithmetic aggregation operators in multiple attribute decision making," *Iranian Journal of Fuzzy Systems*, vol. 13, no. 4, pp. 1–16, 2016.
- [42] D. N. Liu, "Model for evaluating the electrical power system safety with hesitant fuzzy linguistic information," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 2, pp. 725–730, 2015.
- [43] G. Y. Gu, F. J. Wei, and S. H. Zhou, "Risk assessment method for mass unexpected incident in city with hesitant fuzzy linguistic information," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 5, pp. 2299–2304, 2015.
- [44] R. F. Muirhead, "Some methods applicable to identities and inequalities of symmetric algebraic functions of n letters," *Proceedings of the Edinburgh Mathematical Society*, vol. 21, p. 144, 1902.
- [45] P. Liu and D. Li, "Some Muirhead mean operators for intuitionistic fuzzy numbers and their applications to group decision making," *PLoS One*, vol. 12, no. 1, p. e0168767, 2017.
- [46] J. Qin and X. Liu, "2-tuple linguistic Muirhead mean operators for multiple attribute group decision making and its application to supplier selection," *Kybernetes*, vol. 45, no. 1, pp. 2–29, 2016.

- [47] M. Delgado, J. L. Verdegay, and M. A. Vila, "Linguistic decision making models," *International Journal of Intelligent Systems*, vol. 7, no. 5, pp. 479–492, 1991.
- [48] H. Liao, Z. Xu, and M. Xia, "Multiplicative consistency of hesitant fuzzy preference relation and its application in group decision making," *International Journal of Information Technology & Decision Making*, vol. 13, no. 1, pp. 47–76, 2014.
- [49] Z. Xu and M. Xia, "Distance and similarity measures for hesitant fuzzy sets," *Information Sciences*, vol. 181, no. 11, pp. 2128–2138, 2011.

Research Article

Communication Analysis of Network-Centric Warfare via Transformation of System of Systems Model into Integrated System Model Using Neural Network

Bong Gu Kang ¹, Kyung-Min Seo ², and Tag Gon Kim ¹

¹Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea

²Naval & Energy System R&D Institute, Daewoo Shipbuilding & Marine Engineering (DSME) Co., Ltd., Seoul, Republic of Korea

Correspondence should be addressed to Kyung-Min Seo; kmseo.kumsung@gmail.com

Received 11 December 2017; Accepted 11 March 2018; Published 24 June 2018

Academic Editor: Arturo Buscarino

Copyright © 2018 Bong Gu Kang 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.

Communication system in the network-centric warfare (NCW) has been analyzed from the perspective of the system of systems (SoS), which consists of a combat system and a network system so that the two reflect each other's effects. However, this paradoxically causes a prolonged execution time. To solve this problem, this paper proposes an advanced integrated modeling method for the communication analysis in the NCW via the transformation of the SoS, which reduces the simulation execution time while ensuring the accuracy of the communication effects. The proposed models mainly cover interentity traffic and intraentity mobility developed in the form of feed-forward neural networks to guarantee two-way interactions between the combat system and the network system. Because they are characterized as discrete events, the proposed models are designed with the discrete-event system specification (DEVS) formalism. The experimental results show that the proposed transformation reduced an error by 6.40% compared to the existing method and reduced the execution time 3.78-fold compared to the SoS-based NCW simulation.

1. Introduction

Network-centric warfare (NCW) is an emerging theory of war based on the concepts of nonlinearity and complexity [1, 2]. According to the doctrine of NCW, all combat entities, such as war-fighters, manned and unmanned platforms, and command and control (C2) centers, are linked via a network architecture for sharing the combat situation [3, 4]. For example, network nodes are dynamically located as the entities are moved; traffic between the nodes is also changed whenever they interact in joint operations. Thus, NCW should be analyzed based on the communication interactions, which mainly cover interentity traffic and intraentity mobility [5, 6].

For communication analysis in NCW, modeling and simulation (M&S) techniques have been widely used; and a well-known approach is to simulate communication factors separately within the whole NCW simulation [7, 8]. In other words, an M&S engineer separates a network system for explicit communication information from a combat system that generates the overall combat scenario. Then, the

engineer interacts with the two systems under standardized architectures such as high-level architecture (HLA) or test and training enabling architecture [9, 10]. This is a typical approach for system of systems (SoS) development, which supports the high reality of component systems and flexibility in changing them [11, 12].

Despite these advantages, the SoS-based simulation causes a prolonged simulation execution time. Simulation analysis generally requires performing simulation evaluations of all possible input combinations as a “what if” analysis; thus, it consumes a lot of execution time due to the many repeated experiments [13]. In the SoS approach, the simulation overhead due to the use of the interoperation architecture is a primary reason for the time-consuming problem [14, 15].

To overcome this weakness, this paper focuses on a transformation approach that shifts from an SoS-based NCW simulation to an integrated form without using the interoperation infrastructure. For simulation analysis of the network system, we distinguish two main factors: interentity traffic

and intraentity mobility. Based on the factors, we propose (1) a new combat model that includes them and we describe (2) how the model is connected to the network model in the manner of an integrated simulation.

Traffic and mobility, in the integrated simulation, should be two-way interactive with the combat model and the network model. For example, the combat model requests the traffic situation, such as packet delivery ratio and delay, between the source and the destination nodes to the network model. Then, the network model eventually responds to the request based on the network conditions. Thus, the combat model proposed in this paper facilitates the representation of interactive changeability according to the current state of the network; and the network model enables an analysis of the communication effect more realistically. This interactive design for the two models represents a clear difference from previous studies on the development of traffic and mobility models.

To this end, this paper proposes a transformation process consisting of the following three major phases: (1) traffic and mobility data acquisition from SoS-based simulation, (2) data preprocessing for training, and (3) traffic and mobility models hypothesis and variable estimation of them using a neural network. In the third phase, we design regression models in the form of feed-forward neural networks. Specifically, inputs of the models are the communication effects, for example, packet delivery ratio (PDR) and end-to-end delay, and outputs are the variables, for example, interdeparture time of traffic generation and the movement speed of the network node. Because the inputs and the outputs are characterized as discrete events, the traffic and the mobility models are designed with discrete-event system specification (DEVS) formalism [16].

For training of the communication effects, in the first two phases, we preferentially performed SoS-based simulation. The SoS simulation consists of two systems and an interoperable middleware between them: a combat system containing military operations of various combat entities, which is implemented through DEVSim++ [17], a network system that includes the depiction of a mobile ad hoc network (MANET) with network simulator 3 (NS3) [18, 19], and a runtime infrastructure (RTI) for interoperating them [20]. In this paper, we assume that the simulation results for SoS-based NCW are already validated.

As an experiment, we measured the accuracy and simulation speed of the procedure of transformation by comparing the previous traffic and mobility model in the existing study. The experimental result shows that the proposed work reduced the error by about 6.40% compared to the previous work and within an acceptable added time for training the proposed model. Finally, we expect that our study will provide an alternative way for the user, when operating an SoS requiring long execution times, to conduct a simulation analysis of various scenarios including network parameters such as a sensitivity analysis.

This study is organized as follows. Section 2 describes the background. Section 3 analyzes previous work and its limitations. Section 4 defines our problem and explains the proposed method and model. Section 5 discusses the

experimental results by comparing previous studies. Finally, Section 6 presents our conclusions.

2. Background

This section provides background knowledge regarding systematic views and the main factors affecting communication analysis in NCW.

2.1. Two Systematic Views for Communication Analysis in NCW. The power of NCW is derived from the effective linking or networking of knowledgeable entities that are geographically or hierarchically dispersed [21]. The combat entities in NCW can independently move in any direction; also, they are interconnected with wireless communication. Thus, we assume that the network for NCW is realized with MANET, which is effective in enabling highly mobile, highly responsive, and quickly deployable entities [22]. In this context, one of the purposes of NCW simulation is to analyze the performance of the network, for example, packet delivery ratio or end-to-end delay, in consideration of the network's dynamic configuration [23].

Figure 1 shows two systematic views on communication analysis in MANET. The first approach is based on the SoS-based simulation. In the SoS view, a combat system generates the tactical behaviors of all entities and a network system computes the configuration of nodes in MANET. Two main data sets, that is, traffic and mobility, are interacted with each other via a predefined interface. The next view is to develop an integrated network system comprising a network model and an abstracted combat model. The combat model abstracts mobility and traffic data and interacts with the network model at the model level not the system level.

The interoperation method assumes that each subsystem performs its tasks autonomously to take detailed actions for the SoS [11]. Thus, the biggest benefit of the SoS-based NCW is enhancing the accuracy of mobility and traffic data. Nevertheless, it inevitably remains a practical problem due to the prolonged simulation execution time. On the other hand, the integration method implies that each submodel interacts with others, sharing common information to form a unified system. Since they are operated within a standalone environment, the simulation is executed more quickly than in the interoperation method. However, the abstracted models have lower fidelity compared with the independent systems in the SoS.

Therefore, this paper proposes an advanced integration method for communication analysis in NCW, which reduces the simulation execution time and ensures the accuracy of the mobility and traffic models.

2.2. Two Factors for Communication Analysis in NCW: Mobility and Traffic. Figure 2 illustrates how two factors, that is, traffic and mobility, influence the linking of combat entities in NCW. We assume that there is a one-to-one correspondence between a combat entity and a network node. For a straightforward understanding, we explain this by separating

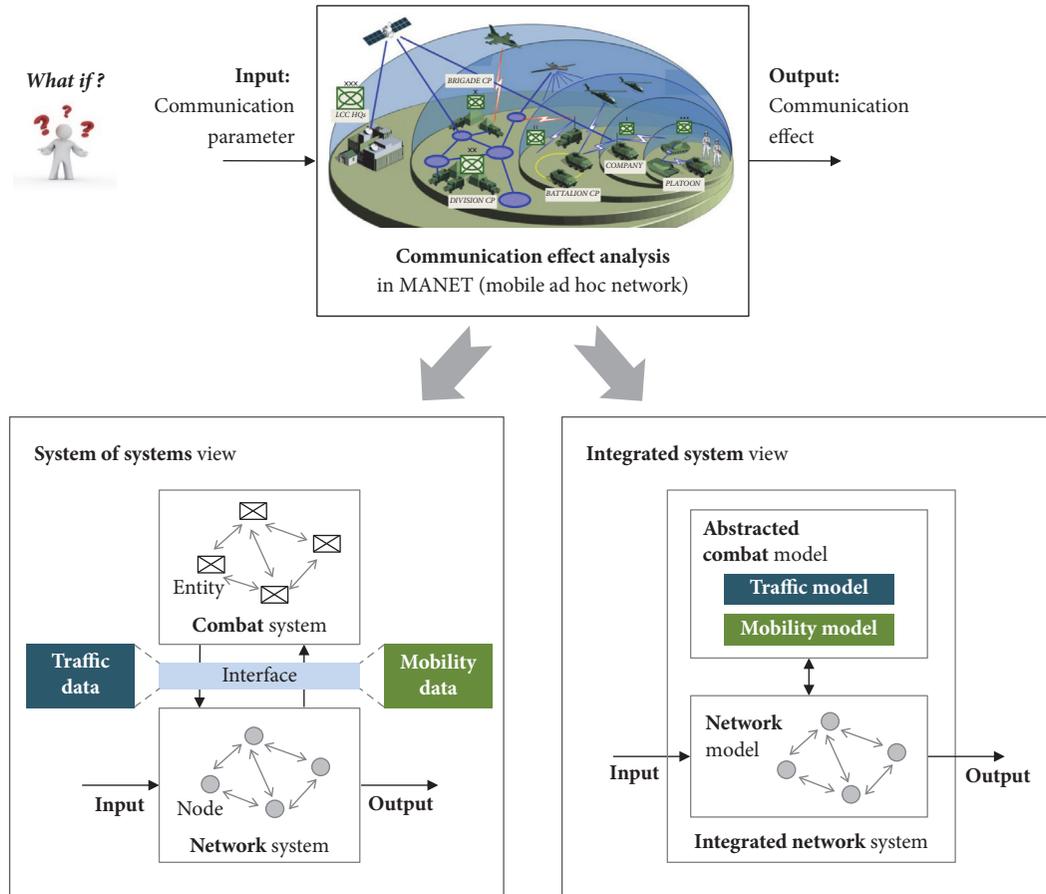


FIGURE 1: Communication analysis from two systematic perspectives: system of systems (SoS) and integrated system views.

the combat entities from their nodes, which is relevant to an SoS approach.

In Figure 2, a C2 center, for example, headquarters, tries to send an infantry troop a command message to move to a specific location. Under normal situations, Entity_{C2} in the combat system sends a request message, including the command to the network system. Then, the network system finds a proper routing path from Node_{C2} to Node_A and sends a response message to Entity_A in the combat system. After receiving the message, Entity_A moves to the desired location. On the contrary, when the network is in a bad condition, the transmission from Node_{C2} to Node_A fails, and the network system cannot send a response to the combat system. In this case, Entity_{C2} has difficulty controlling Entity_A, and Entity_A may generate different maneuvering behavior. If Entity_A moves, whether it behaves correctly or not, it sends a final position to the network system to update its node. In other words, this ad hoc configuration of the network may change with time as the nodes move or adjust their transmission and reception messages.

In this way, the combat system exchanges two types of data, that is, traffic and mobility, with the network via an interoperation infrastructure. The traffic data regarding the command message in Figure 2 is composed of the traffic flow in the connection of the pair of the source and the destination

entities and the requested time for the connection. The mobility data has the position of the entity at the specific time.

With these data, when we transform the SoS-based NCW simulation into the integrated simulation, including the combat and the network models, the following two requirements should be considered. First, traffic and mobility occur eventually in the combat model; thus, the two models are specified with discrete-event simulations. Next, because these data influence both models, it is necessary to construct them with inputs as well as outputs.

Although many research studies have been conducted on the construction of the traffic and mobility model in the network system [24, 25], these regard the models as a generator model with only output. In the following section, we will discuss some related work and its limitations in detail.

3. Related Works and Limitations

Over the last decade, several studies have been conducted to construct a traffic and mobility model for evaluation of the network system in the NCW. Each of these studies has described this model in the form of a generator that can be classified in terms of whether data is used in the process of the model's construction or not. The generator model not based

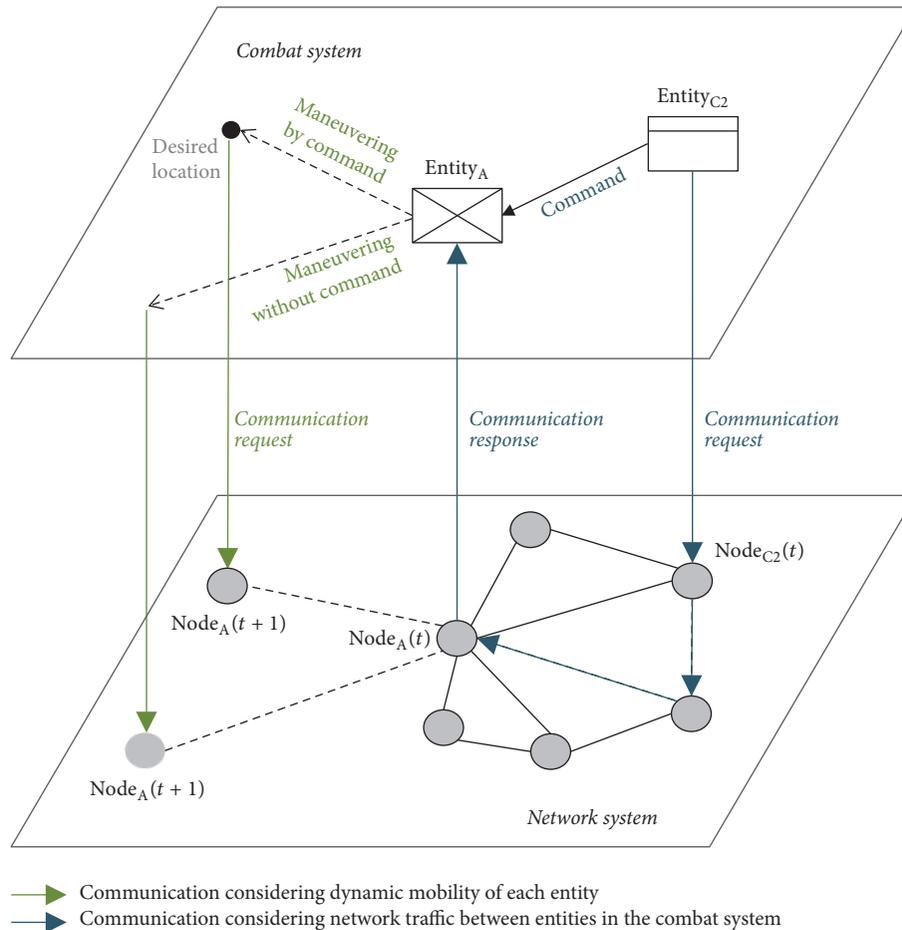


FIGURE 2: Traffic and mobility data exchanged between the combat system and the network system.

on the data has an advantage in that it can be used regardless of the presence or absence of data, whereas it is difficult to reflect the real world completely. On the other hand, the generator model based on data enhances the reality of the traffic and mobility, although it requires data from the real world. Table 1 shows the related traffic and mobility models according to the criteria for this classification.

To evaluate the performance of the network system, even though there is no available data, several studies have used the statistical traffic models based on probabilistic distribution, such as constant bit ratio and exponential. Some of them have represented the network with homogeneous nodes with the same traffic [26–28] or with heterogeneous nodes with different traffic according to the type of the combat entity [29]. On the other hand, other studies have tried to construct the traffic model with available data from the real world [30, 31]. They have extracted the data set of the interdeparture time from the real world and have made the empirical cumulative distribution function using the data set. They have then generated the traffic based on the distribution function. Although these models play a role in generating traffic, they cannot generate the different traffic against a change in the situation, because they are only in the form of a generator with the output.

Similar to the traffic model, some researchers have focused on developing a mobility model to evaluate the network system under a condition in which there is no available data. Bindra et al. and Kioumourtzis et al. used the reference point group mobility model that regards mobile nodes as a moving group based on a random waypoint model (RWP) [32, 33]. Also, Fongen et al. used the hierarchical group mobility model that divides the battlefield area according to the organization structure and moves within the area using the RWP model [34], while Reidt and Wolthuse used the ad hoc mobility model with the tactical maneuver trace predefined by the user [35]. Although their studies can be an alternative when there is no available data, the stereotyped model is somewhat different from the situation in the real world. On the other hand, other studies have tried to construct a similar mobility model that reflects the real world from the data [36, 37]. In these studies, they hypothesized a mobility model that includes the combat entity's property as the parameter, such as speed, the angle of movement, and the duration time of the movement, and then tunes the parameters based on the traced data acquired from the real world. In the end, two such models can only generate the predefined positions regardless of the change of the situation, because they are only generator models.

TABLE I: Related works for construction of the traffic and mobility model.

Model type	Pros and cons	Kinds of traffic and mobility model
Generator model not based on the data	Pros (i) Can be used as an alternative if there is no available traffic or mobility data Cons (i) Is limited in terms of expressiveness	Traffic model (i) Statistical traffic model [26–29] Mobility model (i) Reference point group mobility model [32, 33] (ii) Hierarchical group mobility model [34] (iii) Ad hoc user-defined model [35]
Generator model based on the data	Pros (i) Enhances the reality of the traffic and mobility Cons (i) Requires the traffic or mobility data for learning	Traffic model (i) Traffic model from data in the real world [30, 31] Mobility model (i) Mobility model from data in the real world [36, 37]

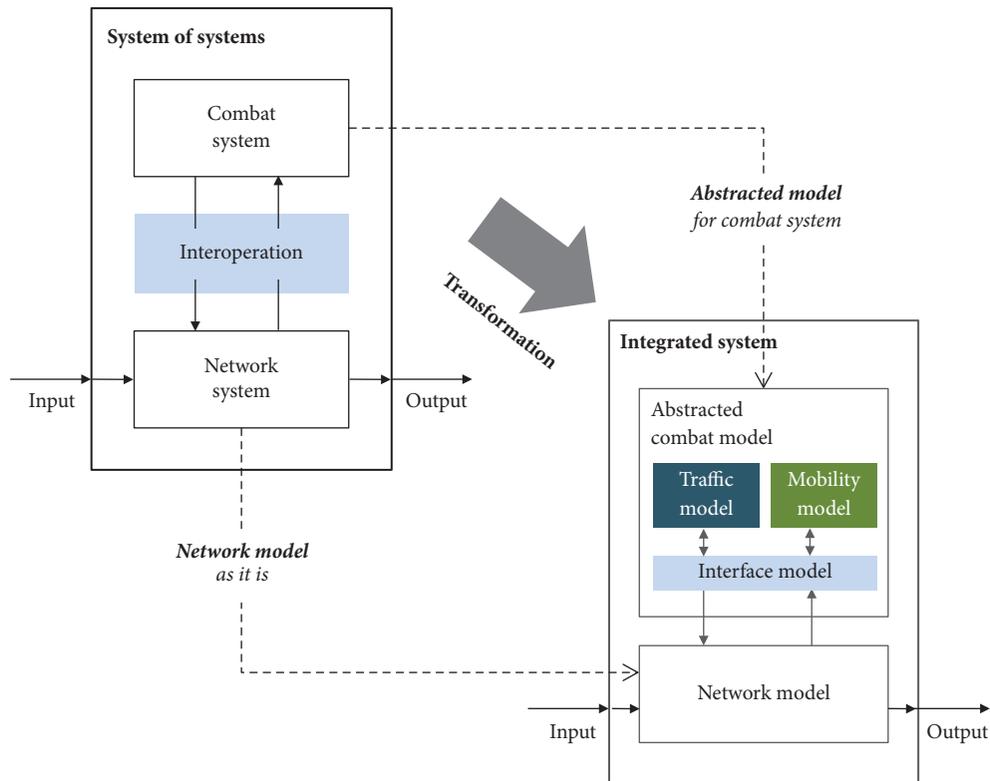


FIGURE 3: Transformation of the SoS into the integrated system for NCW simulation.

To summarize, even though some researchers have tried to construct the traffic and mobility models regardless of the presence or absence of data for the analysis of the NCW, they have represented the models in the form of a conventional generator, only having an output from the view of the standalone system and not the system of systems. Unfortunately, to apply these generator models to the analysis of the network system in the SoS-based NCW simulation causes the degradation of the accuracy, because the traffic and mobility data are changed according to the status of the communication, as we previously explained. For this reason, it is inevitable that we will need to construct a new type of traffic and mobility model that includes the input and output to generate the different traffic and mobility traces according to the status of the network system. Therefore, we suggest a

new type of traffic and mobility model that satisfies the above constraints of the previous studies. The following section focuses on this proposed model.

4. Proposed Work

In this section, we clarify the problem definition and propose a transformation method for NCW simulation. The proposed model in the method is an NCW model including an abstracted combat model and network model.

4.1. Problem Definition. Figure 3 shows how this study transforms the SoS based on an interoperation environment (upper left part) to the integrated system (lower right part). In this process, the network system remains as it is, whereas

the combat system is abstracted into an abstracted combat model, which includes the traffic and mobility generating the traffic and position data, and an interface model connecting the models with the network model. Then, the model is integrated with the unchanged network model.

In the process of the abstraction of the combat system, to deal with the aforementioned limitations, the proposed abstracted combat model should satisfy two requirements: (1) a discrete-event model for discrete-event simulation and (2) a model generating the different output according to the states of the adjacent system. In this respect, we define such a situation as a problem to (1) hypothesize a DEVS coupled model including atomic models and (2) estimate the related variables.

4.2. Proposed Transformation Method. In general, the output of the system ($Y = f(X; V)$) can be expressed as a function (f) of input, that is, parameters (X) and variables (V). From this perspective, Figure 4 shows how formalism expresses the transformation, where

f is the combat system to be abstracted

g is the network system to be analyzed

X_g is the input parameter set of g ;

Y_g is the output set of g ;

V_g is the variable set of g ;

Y_{fg} is the output set from f to g ;

Y_{gf} is the output set from g to f ;

\widehat{f}_g is the abstracted model of f from the perspective of g ;

V_{fg} is the variable set of \widehat{f}_g ;

\widehat{h} is the estimation function for V_{fg} ;

$\widehat{Y}_{fg} = \widehat{f}_g(\widehat{Y}_{gf}; V_{fg})$ is the estimated output from \widehat{f}_g ;

$V_{fg} = \widehat{h}(\widehat{Y}_{gf})$ is the estimated variable of the model (\widehat{f}_g);

$\widehat{Y}_g = g(X_g, \widehat{f}_g(\widehat{Y}_{gf}; \widehat{h}(\widehat{Y}_{gf})); V_g)$ is the integrated system.

In the SoS, the output of the network system is $Y_g = g(X_g, Y_{fg}; V_g)$, and Y_{fg} is acquired from the combat system ($Y_{fg} = f(Y_{gf}; V_f)$). Unfortunately, because the combat system does not exist in the integrated system, Y_{fg} cannot be acquired directly.

To solve this problem, we hypothesize an abstracted model ($\widehat{Y}_{fg} = \widehat{f}_g(\widehat{Y}_{gf}; V_{fg})$). \widehat{f}_g and V_{fg} refer to an abstracted combat model and a variable of the combat model from the perspective of the analysis of the network system. For example, among the various combat system logics, factors only related to the traffic and position are necessary for

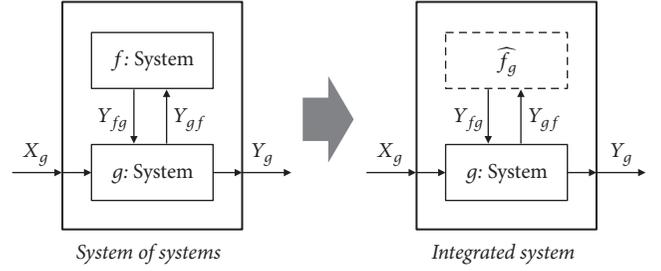


FIGURE 4: Systematic representation of the proposed transformation.

the network system's analysis. In this context, V_{fg} means the variable for expressing the factors and \widehat{f}_g means the abstracted combat model, including the variable. In addition, to represent the fact that V_{fg} is influenced by the response of the network system (i.e., \widehat{Y}_{gf}), we assume an estimation function (\widehat{h}) against the relation between V_{fg} and \widehat{Y}_{gf} ; that is, $V_{fg} = \widehat{h}(\widehat{Y}_{gf})$.

With \widehat{f}_g and \widehat{h} , this paper can acquire the output, $\widehat{Y}_{fg} = \widehat{f}_g(Y_{gf}; \widehat{h}(\widehat{Y}_{gf}))$, from the abstracted combat model, and furthermore, by using this model, we can finally find the output of the integrated network system $\widehat{Y}_g = g(X_g, \widehat{f}_g(\widehat{Y}_{gf}; \widehat{h}(\widehat{Y}_{gf})); V_g)$ by integrating the network model. The next subsection shows its detailed process.

4.3. Overall Process of System Transformation. Figure 5 depicts the process of the transformation consisting of the following three major phases: (1) data acquisition from the simulation of the SoS-based NCW, (2) data preprocessing, and (3) traffic and mobility model hypothesis and variable estimation of the models using the neural network. In the model hypothesis step, Figure 6 and the following specification indicate an abstracted combat model (ACM) constructed as a DEVS coupled model, including three kinds of DEVS atomic models: traffic (TM), mobility (MM), and interface model (IM). The role of the traffic model is to generate data from the source node to the destination node; each traffic model is mapped onto the node's pair of the source node and the destination node. The mobility model generates the position of the corresponding node; each model is mapped to each node.

In the last phase, we hypothesized the traffic and mobility model and the function for estimating the variables of the models using the neural network; we trained the neural network using the preprocessed data set. Then we constructed the integrated system for the NCW by embedding the neural network in the traffic and mobility model. The next subsection gives a description of the last phase.

4.4. Proposed NCW Model. In the first phase, we extracted some experimental points against the input of the network system (i.e., communication parameters) from the entire design space using the design of the experiment (DOE). We

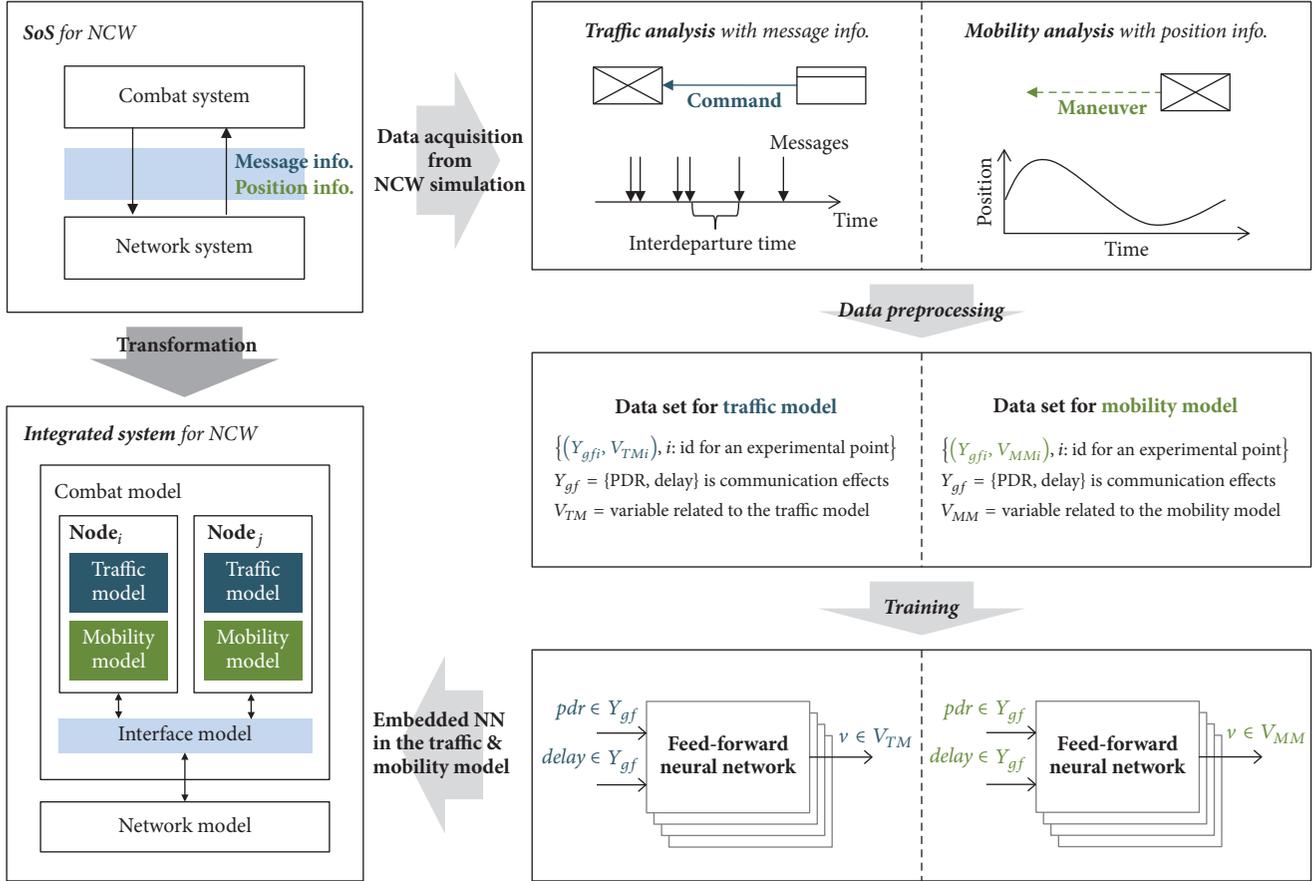


FIGURE 5: Proposed transformation procedure of the SoS into the integrated system.

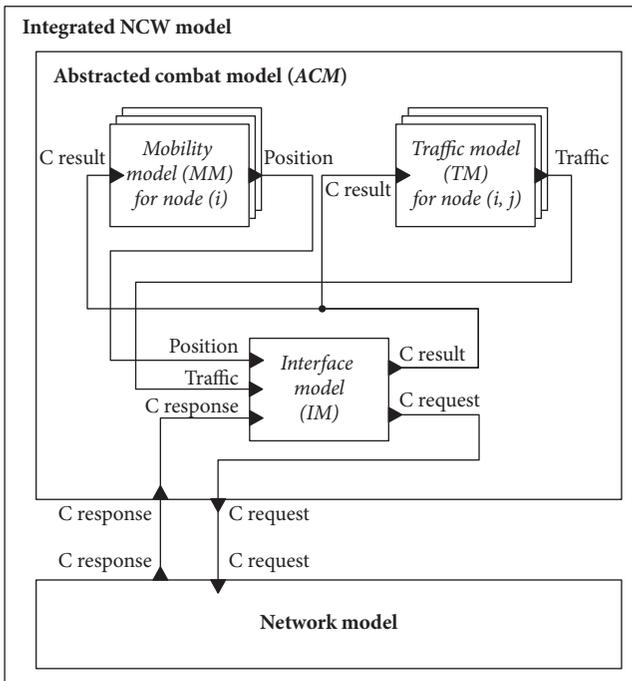
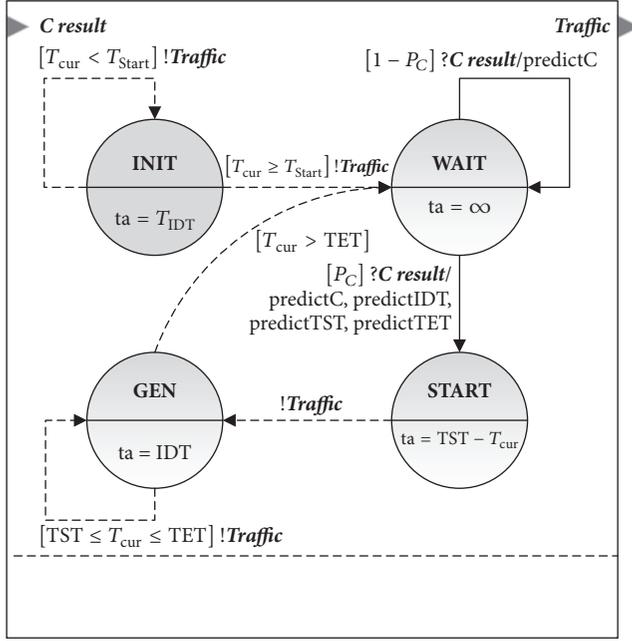


FIGURE 6: Overall NCW model structure including the abstracted combat model and the network model.

then executed the NCW simulation against the extracted experimental points by acquiring the message and position information for the traffic and mobility analysis. After that, for the preprocessing of the data for training, we constructed the data set for traffic and mobility from the acquired information against the experimental points. The former is $\{(Y_{gfi}, V_{TMi}), i: \text{id for an experimental point}\}$ and the latter is $\{(Y_{gfi}, V_{MMi}), i: \text{id for an experimental point}\}$, where Y_{gfi} is communication response, that is, end-to-end delay and PDR, and V_{TMi} and V_{MMi} are the variables related to the traffic and mobility model, respectively.

In addition, to represent the property of the traffic and mobility affected by the network system, the interface model performs a role to connect the communication effects from the network model to the input of the traffic and mobility model according to the following procedure. The interface model transmits the traffic generated from the traffic model to the network model through the C request during the initial period of the simulation. The model then receives the communication effect of each event from the network model through the C response port, calculates the average communication effect, and transmits it to the traffic and mobility model using the C result port. Once it has received the effects, the traffic and mobility model completes the simulation model by calculating the value of the variables of the model based on the effects



→ External transition, Transition Syntax: [Condition] ?input/
 Object model
 - - - Internal transition, Transition Syntax: [Condition] !output/
 Object model

FIGURE 7: Traffic model design using the DEVS formalism.

and the neural network and proceeds with the simulation.

$$ACM = \langle X, Y, \{M_i\}, EIC, EOC, IC, sel \rangle, \quad (1)$$

where

$$\begin{aligned}
 X &= \{C \text{ response}\}; \\
 Y &= \{C \text{ request}\} \\
 \{M_i\} &= \{\bigcup_{i=1}^n TM_i, \bigcup_{i=1}^n MM_i, IM\}; \\
 EIC &= \{(ACM.C \text{ response}, IM.C \text{ response})\}; \\
 EOC &= \{(IM.C \text{ result}, ACM.C \text{ request})\}; \\
 IC &= \{(IM.C \text{ result}, \bigcup_{i=1}^n TM_i.C \text{ result}) \\
 &\quad (IM.C \text{ result}, \bigcup_{i=1}^n MM_i.C \text{ result}), \\
 &\quad (\bigcup_{i=1}^n TM_i.Traffic, IM.Traffic) \\
 &\quad (\bigcup_{i=1}^n MM_i.Position, IM.Position)\}; \\
 sel &= IM.
 \end{aligned}$$

Figure 7 and the following specification show the structure of the traffic DEVS atomic model (TM) consisting of four variables: $V_{TM} = \{IDT, P_C, TST, TET\}$. IDT and P_C refer to the time interval between the generation of the traffic and whether or not the source and destination node are connected. TST and TET are the times when the packet generation is started and ended.

$$TM = \langle X, Y, S, \delta_{ext}^p, \delta_{int}, \lambda, ta \rangle, \quad (2)$$

where

$$X = \{C \text{ result}\};$$

$$Y = \{Traffic\}$$

$$S = \{INIT, WAIT, START, GEN\}$$

$$\delta_{ext}^p : (WAIT, C \text{ result}, P_C) \rightarrow START$$

$$\begin{aligned}
 &\text{execute } \forall om \in OM_{TM}; \\
 &(WAIT, C \text{ result}, 1 - P_C) \rightarrow WAIT \\
 &\text{execute } predictP_C;
 \end{aligned}$$

$$\delta_{int} : (START, TST - T_{cur}) \rightarrow GEN$$

$$(GEN, IDT) \rightarrow GEN \text{ where } TST \leq T_{cur} \leq TET$$

$$(GEN, IDT) \rightarrow WAIT; \text{ where } TET < T_{cur}$$

$$(INIT, T_{IDT}) \rightarrow INIT; \text{ where } T_{cur} < T_{START}$$

$$(INIT, T_{IDT}) \rightarrow WAIT; \text{ where } T_{START} \leq T_{cur}$$

$$\lambda : \{START, GEN, INIT\} \rightarrow Traffic$$

$$ta : INIT \rightarrow T_{IDT};$$

$$WAIT \rightarrow \infty;$$

$$START \rightarrow TST - T_{cur}$$

$$GEN \rightarrow IDT.$$

$$V_{TM} = \{IDT, P_C, TST, TET\} \quad (3)$$

is TM 's variable:

IDT is the interdeparture time of traffic generation;

P_C is the probability of existence of the traffic model;

TST is the start time of the traffic generation;

TET is the end time of the traffic generation.

$$OM_{TM} = \{predictIDT, predictP_C, predictTST, predictTET\} \quad (4)$$

is V_{TM} 's object model:

$predictIDT$ is the prediction function of IDT ;

$predictP_C$ is the prediction function of P_C ;

$predictTST$ is the prediction function of TST ;

$predictTET$ is the prediction function of TET ;

T_{START} is the time for the initial traffic generation;

T_{IDT} is the interdeparture time of the traffic in the $INIT$ state.

The traffic model is a probabilistic discrete-event model in which packets are generated at intervals of IDT from the TST to the TET time according to the probability of the existence of the connection (P_C) [38]. The model has four states: $INIT$, $WAIT$, $START$, and GEN . In the $INIT$ state, traffic occurs in the T_{IDT} interval time during

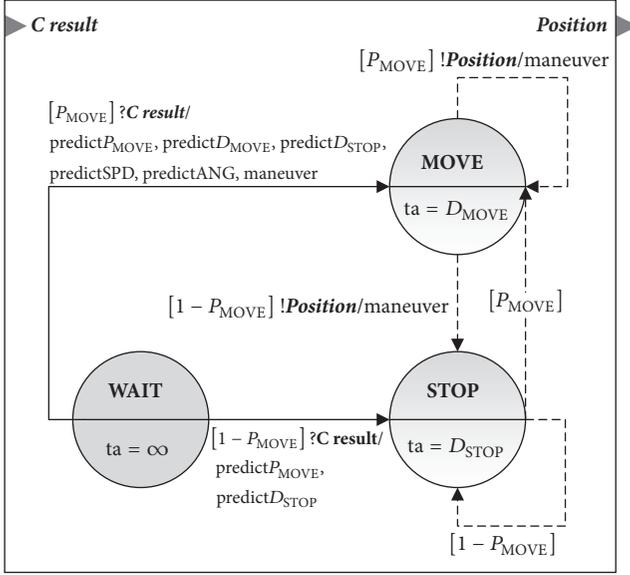


FIGURE 8: Mobility model design using the DEVS formalism.

the T_{START} time and then waits in the *WAIT* state. After receiving the communication effect from the interface model through the *C result*, the IDT , P_C , TST , and TET values are calculated by calling the object model: $OM_{TM} = \{predictIDT, predictP_C, predictTST, predictTET\}$. If the connection exists, it transits to the *START* state and to the *GEN* state after the $TST - T_{cur}$ time for traffic generation and generates the traffic at the IDT interval. Then, when the TET time is reached, the traffic generation is stopped and the state transits to the *WAIT* state. On the other hand, if *C* does not exist, it does not generate traffic. In such a form, this traffic model plays the role of generating different types of traffic according to the communication effect calculated in the network model.

Figure 8 and the following specification show the structure of the mobility DEVS atomic model (*MM*) consisting of five variables: $V_{MM} = \{P_{MOVE}, D_{MOVE}, D_{STOP}, SPD, ANG\}$. P_{MOVE} refers to the transition probability to the *MOVE* state, and D_{MOVE} and D_{STOP} refer to the time for remaining in the *MOVE* and *STOP* state, respectively. In this model, the probability of staying in the *MOVE* state is determined by P_{MOVE} , and the position information is generated based on the SPD and ANG in that state.

$$MM = \langle X, Y, S, \delta_{ext}^p, \delta_{int}^p, \lambda, ta \rangle, \quad (5)$$

where

$$X = \{C \text{ result}\}$$

$$Y = \{Position\};$$

$$S = \{WAIT, MOVE, STOP\}$$

$$\delta_{ext}^p : (WAIT, C \text{ result}, P_{MOVE}) \rightarrow MOVE$$

$$\begin{aligned} & \text{execute } \forall om \in OM_{MM} \\ & (WAIT, C \text{ result}, 1 - P_{MOVE}) \rightarrow STOP \end{aligned}$$

execute $predictP_{MOVE}, predictD_{STOP}$

$$\delta_{int}^p : (MOVE, D_{MOVE}, P_{MOVE}) \rightarrow MOVE$$

execute *maneuver*

$$(MOVE, D_{MOVE}, 1 - P_{MOVE}) \rightarrow STOP$$

execute *maneuver*

$$(STOP, D_{STOP}, P_{MOVE}) \rightarrow MOVE$$

$$(STOP, D_{STOP}, 1 - P_{MOVE}) \rightarrow STOP$$

$$\lambda : MOVE \rightarrow Position;$$

$$ta : WAIT \rightarrow \infty;$$

$$MOVE \rightarrow D_{MOVE};$$

$$STOP \rightarrow D_{STOP}.$$

$$V_{MM} = \{P_{MOVE}, D_{MOVE}, D_{STOP}, SPD, ANG\} \quad (6)$$

is *MM*'s variable:

P_{MOVE} is the transition probability to the *MOVE* state;

D_{MOVE} is the duration time in the *MOVE* state;

D_{STOP} is the duration time in the *STOP* state;

SPD is the speed of the movement;

ANG is the angle of direction change.

$$\begin{aligned} OM_{MM} = \{ & predictP_{MOVE}, predictD_{MOVE}, \\ & predictD_{STOP}, predictSPD, predictANG, \\ & maneuver \} \end{aligned} \quad (7)$$

is V_{MM} 's object model:

$predictP_{MOVE}$ is the prediction function of P_{MOVE} ;

$predictD_{MOVE}$ is the prediction function of D_{MOVE} ;

$predictD_{STOP}$ is the prediction function of D_{STOP} ;

$predictSPD$ is the prediction function of SPD ;

$predictANG$ is the prediction function of ANG ;

maneuver is the maneuver function using SPD, ANG .

This model is a probabilistic discrete-event model in which the model varies stochastically according to the P_{MOVE} . The model has three states, *WAIT*, *MOVE*, and *STOP*, and waits initially in the *WAIT* state. Similar to the traffic model, after receiving input through the *C result*, five parameter values are calculated by the object model: $OM_{MM} = \{predictP_{MOVE}, predictD_{MOVE}, predictD_{STOP}, predictSPD, predictANG, maneuver\}$. In the *MOVE* state, the position is updated through the SPD and ANG during the D_{MOVE} time, but the position is not updated during the D_{STOP} time in the *STOP* state. At the end of each state, it decides the probability of staying in the *MOVE* state according to P_{MOVE} .

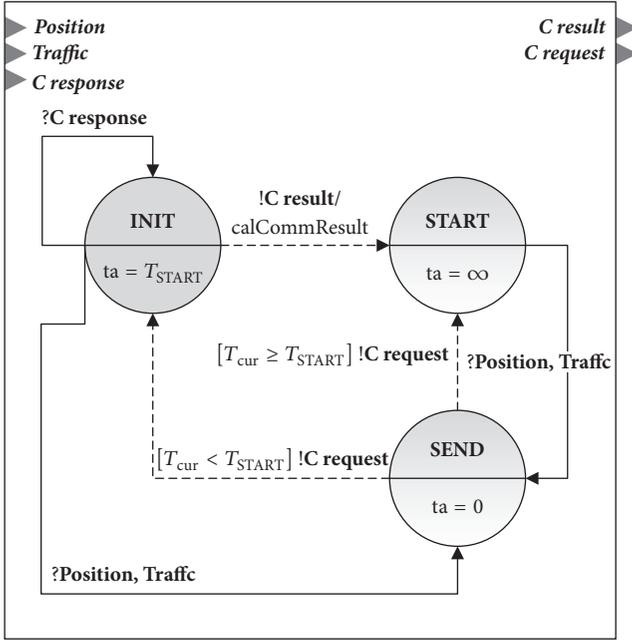


FIGURE 9: Interface model design using the DEVS formalism.

In this way, the influence of the network model affects the values of the five variables through the object model, thereby generating different types of mobility.

Finally, the specification for the interface DEVS atomic model (IM) is described in Figure 9. This model calculates the cumulative communication effect through the C response from the network model during the initial T_{START} time and passes the result to the traffic and mobility model through the C result port. Then, the traffic and position information generated from each model is transmitted to the network model through the C request port. The interface model consists of three states: $INIT$, $START$, and $SEND$. After the T_{START} time in the $INIT$ state, the model calculates the average communication effects using the $calCommResult$ and transmits them to the traffic and mobility model. Through this process, the model enables the generation of the different types of traffic and mobility data according to the state of the network model.

$$IM = \langle X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta \rangle, \quad (8)$$

where

$$X = \{Position, Traffic, C \text{ response}\}$$

$$Y = \{C \text{ result}, C \text{ request}\};$$

$$S = \{INIT, START, SEND\}$$

$$\delta_{ext} : (INIT, C \text{ response}) \rightarrow INIT$$

$$(INIT, Position \text{ or } Traffic) \rightarrow SEND$$

$$(START, Position \text{ or } Traffic) \rightarrow SEND$$

$$\delta_{int} : INIT \rightarrow START;$$

$$SEND \rightarrow INIT \text{ where } T_{cur} < T_{START}$$

$$SEND \rightarrow START \text{ where } T_{START} \leq T_{cur}$$

$$\lambda : INIT \rightarrow C \text{ result};$$

execute $calCommResult$

$$SEND \rightarrow C \text{ request};$$

$$ta : INIT \rightarrow T_{START};$$

$$START \rightarrow \infty;$$

$$SEND \rightarrow 0.$$

$calCommResult$ is the function for calculation on the effects of communication.

After the model hypothesis step, we are going to focus on the variable (V_{TM}, V_{MM}) estimation step. As mentioned in the previous subsection, it is necessary to transform raw data $\{Y_{gf}, Y_{fg}\}$ from the SoS-based NCW simulation to the data set $\{Y_{gf}, V_{fg}\}$ for learning (referring to Figure 5). We then constructed a regression model with the form of feed-forward neural network using the data set [39–41].

The upper part of Figure 10 shows the detailed structure including Y_{gf} as an input and V_{fg} as an output. It consists of three kinds of layers: input, hidden, and output; each layer has neurons, and they have a weighted connectivity with neurons in the other layers. The value of neurons in the hidden and output layer is acquired from the weighted sum of neurons in the previous layer and the activation function. To make this identified regression model in an executable form, as shown in the lower part of Figure 10, this paper regards the regression model as a function and generates the model to the source codes, which can be implemented to the prediction functions of the object model (OM_{TM}, OM_{MM}) in the above traffic and mobility model.

5. Case Study: Simulation-Based Analysis of the Network System

The objective of this case study is to demonstrate how much the proposed method improves the accuracy compared to the existing method while conducting the transformation of the SoS-based NCW [42, 43]. Also, this paper compares the simulation speeds before and after the transformation.

5.1. Experimental Design. The objective of the SoS-based NCW simulation is to analyze the communication effects (i.e., end-to-end delay and PDR) against the communication-related parameters under the battlefield environment in which the exchange of information between combat entities occurs through the communication. The SoS-based NCW consists of two systems (combat and network system), which have already been validated.

The combat system represents the army's military logic at an infantry company level and consists of 131 combat entities. These entities exchange information between entities through communication (traffic) and conduct the maneuver (mobility). This system is implemented in the DEVSIMHLA using DEVS formalism. On the other hand, the network

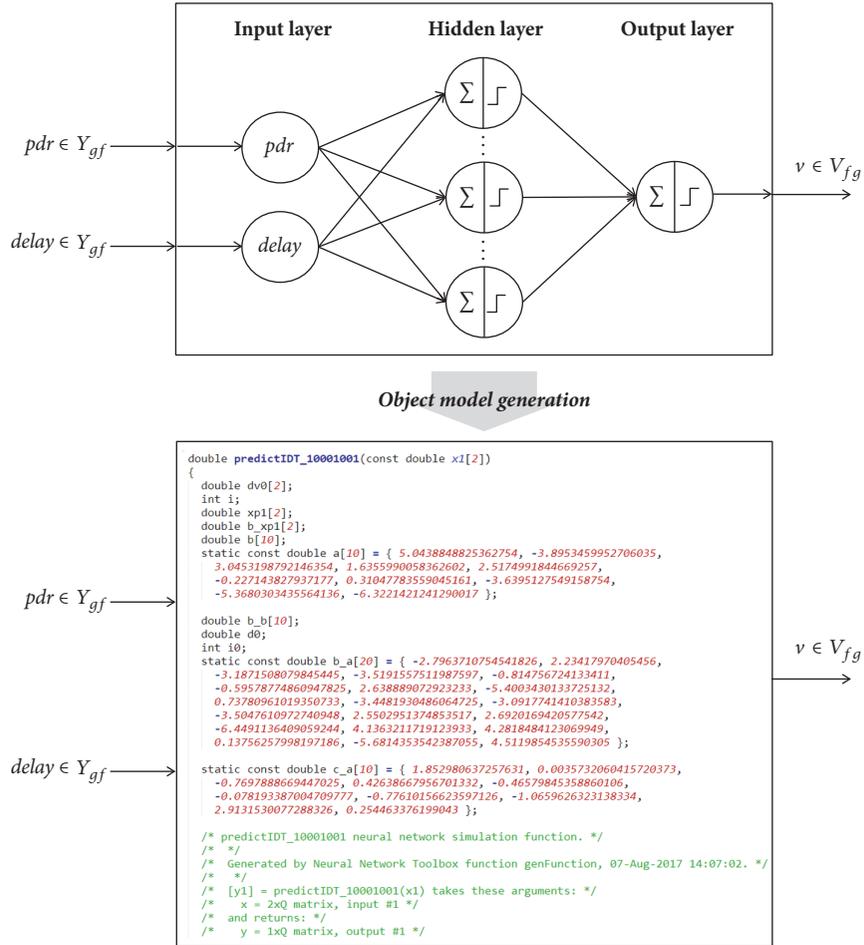


FIGURE 10: Realization of the object model from the trained feed-forward neural network.

TABLE 2: Network system parameters and their description.

Parameter name	Parameter level	Description
Packet size (P_{PS})	100, 200, 400, ..., 6400 (byte)	The size of packet
Transmission power (P_{TP})	-10, 5, 0, ..., 40 (dbm)	The transmission power of node
Transmission gain (P_{TG})	0, 2, 4, ..., 20 (dB)	The transmission gain of node
Reception gain (P_{RG})	0, 2, 4, ..., 20 (dB)	The reception gain of node
PhyMode (P_{PM})	1, 2, 5.5, 11 (Mbps)	The 802.11 phy layer mode of DsssRate

system describes the MANET using destination sequenced distance vector (DSDV) routing protocol and consists of 131 network nodes, which correspond to the entities of the combat system [44, 45]. This system calculates the effects of communication (traffic) by updating the position of nodes (mobility). This system is implemented in the ns-3 discrete-event network simulator.

The two systems participate in the HLA-based interoperable simulation through an RTI and use a federation object model that includes the traffic and mobility data. In addition to the data management, the systems advance the simulation time using the application program interfaces related to time management. The network system calculates the effects of communication against the traffic data from the

combat system and transmits them to the destination node by updating information on the position from the mobility data.

This combat scenario depicts the complex and hierarchical information exchange among combat entities [46, 47]. The blue force conducts a defense operation against the red force with three times the military strength in a 2 km \times 2 km operation area. The combat entities perform threat evaluation and weapons assignment and transmit the results to another entity through communication. According to the communication performance, the entity makes a different decision and generates a different type of traffic and mobility data. Table 2 shows the parameters' names and their descriptions.

TABLE 3: Linear regression model of packet delivery ratio.

	Coefficient	SE	t Stat	p value
C	0.3844	0.0498	7.7202	$1.63e - 10$
P_{PS}	$-1.59e - 05$	$7.00e - 06$	-2.2756	0.0265
P_{TP}	0.0102	$9.4723e - 04$	10.7808	$1.43e - 15$
P_{TG}	0.0088	0.0024	3.7153	$4.5282e - 04$
P_{RG}	0.0084	0.0024	3.5254	$8.2480e - 04$
P_{PM}	-0.0160	0.0043	$-3.6865e - 04$	$4.9648e - 04$

Adj. $R^2 = 0.712$; p value = $8.14e - 16$.

TABLE 4: Linear regression model of end-to-end delay.

	Coefficient	SE	t Stat	p value
C	0.0775	0.0660	1.1737	0.2452
P_{PS}	$4.68e - 05$	$9.28e - 06$	5.0439	$4.64e - 06$
P_{TP}	0.0033	0.0013	2.6683	0.0098
P_{TG}	$9.3956e - 04$	0.0031	0.2987	0.7662
P_{RG}	$6.5681e - 04$	0.0031	0.2088	0.8354
P_{PM}	-0.0234	0.0057	-4.0851	$1.34749e - 04$

Adj. $R^2 = 0.39$; p value = $1.66e - 06$.

5.2. Experimental Procedure. Before conducting the transformation, we executed the SoS-based NCW simulation. In our five-dimensional network parameters in Table 2, the full factorial design size is 7 (for P_{PS}) \times 11 (for P_{TP}) \times 11 (for P_{TG}) \times 11 (for P_{RG}) \times 4 (for P_{PM}) = 37,268, which requires a long execution time, approximately 1,128,084 hours if we conduct 30 trials against each experimental point (1.01 hours per one execution) [48–50]. For this reason, we selected the first 65 experimental points for training from the entire design space: 43 points using face-centered central composite and 22 points using a Latin hypercube design [51]. We then chose an extra 22 points at random from among the full design to evaluate the transformation. Using the acquired data from the SoS-based NCW simulation, we constructed an abstracted combat model consisting of the 908 traffic models, 131 mobility models, and an interface model. In the process, we trained the feed-forward neural network with a 2-5-1 structure using Levenberg-Marquardt algorithm.

From the perspective of accuracy and simulation execution performance, we regarded the output of the network system and the execution time as the effectiveness index. The environment for this case study is as follows. For the combat system, CPU: I5-3550 3.3 GHz, RAM: 4 GB, DEVSim++ v.3.1 are used. For the network system, we used NS3 v.3.18. These two systems used RTI 1.3-NG and progressed simulation time over 5,000 sec including T_{START} 300 sec. For the training, we used MATLAB neural network toolbox v.8.2.1 and MATLAB Coder v. 2.7.

5.3. Experimental Results. Before the analysis of the accuracy and the speed, we constructed the first-order linear regression model ($y \sim 1 + P_{PS} + P_{TP} + P_{TG} + P_{RG} + P_{PM}$) to identify whether the selected network parameters influence the communication effects [52, 53]. In Tables 3 and 4, the column refers to the parameters, while the row refers to

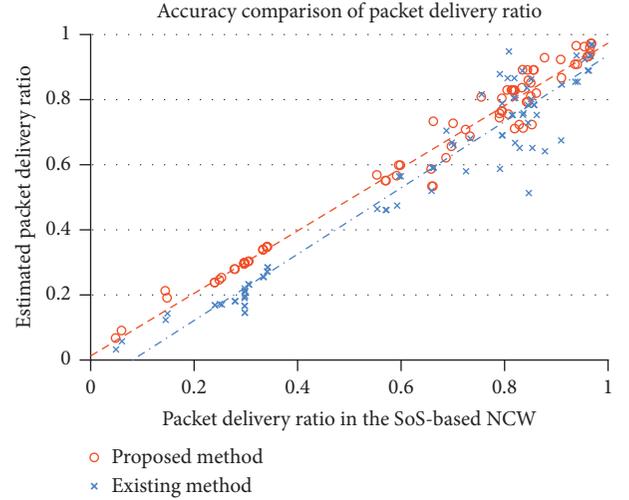


FIGURE 11: Simulation results for packet delivery ratio comparing between the existing and the proposed methods.

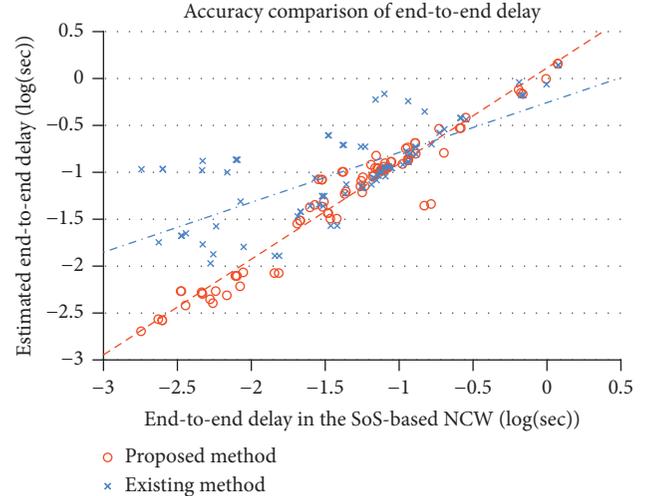


FIGURE 12: Simulation results for end-to-end delay comparing between the existing and the proposed methods.

the coefficient estimates, the standard errors (SEs) of the estimates, the t -statistic values of the hypothesis tests for the corresponding coefficients (t Stat), and the significant probability (p value). Tables 3 and 4 show that the five parameters influence at least one of the two communication effects based on the fact that the p value is smaller than the significance level (0.05); therefore, we used these parameters in the case study.

To conduct an analysis from the perspective of accuracy, we compared the results of the transformation through the proposed method and the conventional method, which only generates the output regardless of the input. Figures 11 and 12 show the graphs for the accuracy comparison of the communication effects (i.e., PDR and end-to-end delay). The x -axis and y -axis refer to the communication effects from the SoS-based NCW execution and integrated system, respectively. The more symmetry of the x - and y -axes, the

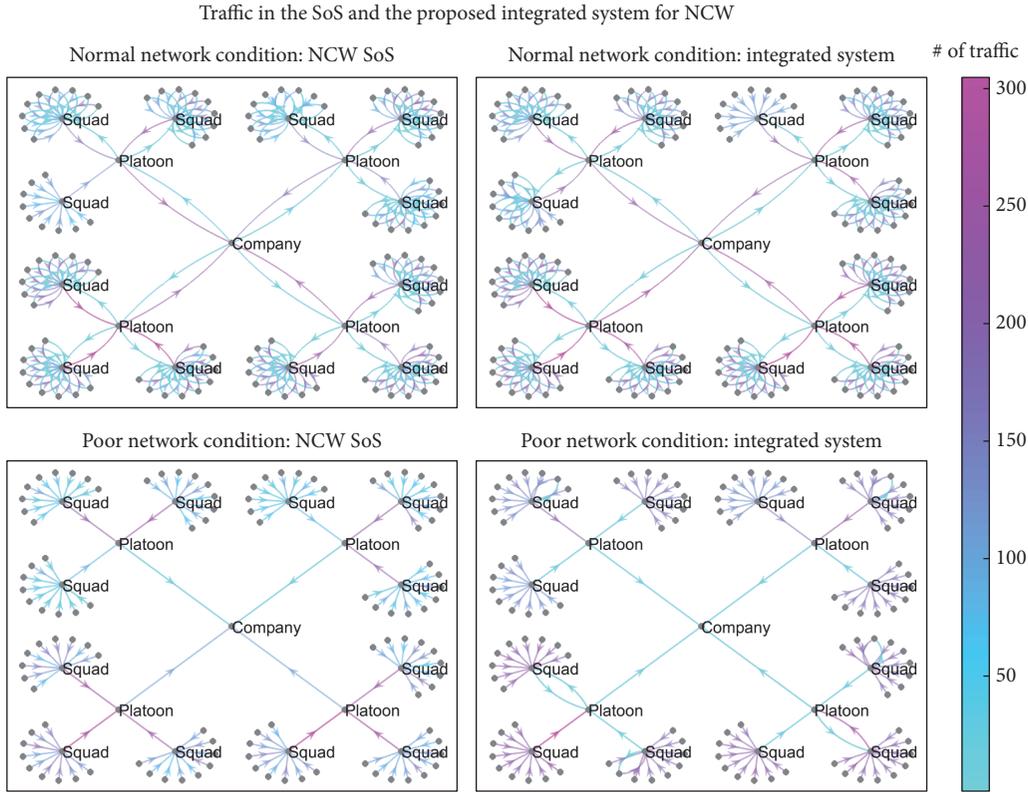


FIGURE 13: Traffic model results of the SoS and the proposed integrated system for NCW.

higher the accuracy. In Figures 11 and 12, the simulation results of the proposed method have higher accuracy than those of the conventional method. For the quantitate analysis of this difference, we measured the root mean square error (RMSE). In the case of Figure 11, each case indicates 0.0425 and 0.0997 RMSE, which means 4.6281% and 10.8407% when considering the minimum and maximum value (0.0491, 0.9691). Also, Figure 12 indicates 0.0520- and 0.1304-second RMSE, which means 4.3721 and 10.9647% when considering the minimum and maximum value (0.0018, 1.1915). This improved accuracy stems from the proposed traffic and mobility model, as shown in the following figures.

Figure 13 shows the flow of the traffic, which is one of the causes of enhanced accuracy. The left and right part show the traffic between entities in the SoS-based NCW and the proposed integrated system. The upper and lower part show the traffic in the normal and poor network condition; the former implies a case of having high PDR and small end-to-end delay, and the latter implies a case of having small PDR and high end-to-end delay. In the left part, the two figures show that more traffic occurs in the normal communication condition than in the poor condition, because the normal network condition allows for more exchanged information and more connection of information owing to the delivery of the hierarchical command. From this perspective, the right part describes a similar trend to the left part in that traffic changes according to the communication condition, which helps to reduce the error from the transformation.

TABLE 5: Comparison of the simulation speeds of the SoS and the integrated system for NCW.

	Execution time (min.)	# of processed events in the network system
SoS	60.539	2.539e + 08
Integrated system	16.025	2.569e + 08

Figure 14 shows another cause of enhanced accuracy, mobility data; it shows the average position change of the nodes against the experimental points at the end of the simulation time. In the SoS-based NCW simulation, the position change in the experimental points with a normal communication condition is larger than the experimental points with a poor condition, as the orders between the entities related to maneuver are normally transmitted through the communication. In the proposed integration system, the mobility also indicates a similar trend, although it does not have the same results; it also plays a role in enhancing the accuracy.

From the perspective of the simulation speed, we compared the average execution times and the average numbers of executed events per one trial in the SoS-based NCW and integrated system. Table 5 shows that the integrated system reduced the execution time 3.78-fold compared to the SoS-based NCW. Also, judging by the fact that the

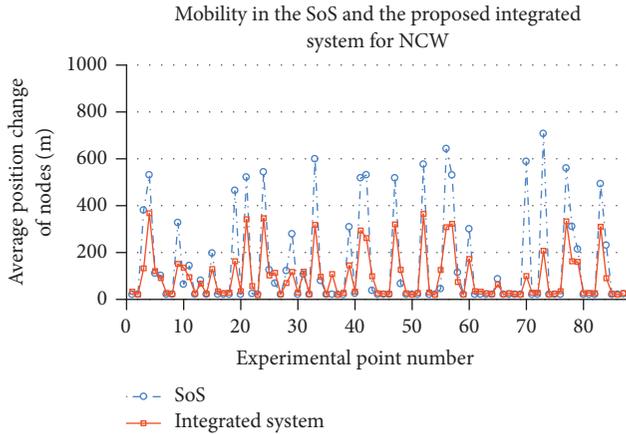


FIGURE 14: Mobility model results of the SoS and the proposed integrated system for NCW.

number of executed events in the network system is similar between the SoS-based NCW and the integrated system and is much higher than the number of the executed events in the combat system of the SoS-based NCW, we can infer that the elimination of the interoperation architecture of the SoS-based NCW plays a prominent role in reducing the execution time.

Furthermore, they recorded 2633.45, 697.07 hours with the 87 experiment points, that is, 2610 trials. In addition to the execution time for the simulation, the proposed integrated system requires a time for training in machine learning of about 8.55 hours, including 7.27 hours for the traffic models and 1.28 hours for the mobility models. Fortunately, however, the time is quite small compared to the execution time of the SoS-based NCW simulation.

6. Conclusion

In network-centric warfare (NCW), due to the importance of communication, which is responsible for the flow of information, it is necessary to analyze the performance of this communication against the communication parameters in an environment with high complexity, such as a battlefield. For this reason, many studies have conducted a simulation-based analysis of the NCW from the perspective of the SoS, which consists of the combat system and the network system so that the two reflect each other's effects. However, this paradoxically causes a prolonged execution time and difficulty in conducting the analysis of the various parameters due to the problem of time.

To overcome this weakness, we need to abstract the combat system to an abstracted combat model that includes the traffic and mobility models required for the network system's analysis. We also need to integrate the abstracted combat model with the network system. Some studies have been conducted on the construction of the traffic and mobility model for the analysis of communication. However, as the analysis has been performed in a standalone system, not an SoS, the resulting model has a form with only an output,

and therefore the model cannot generate the different outputs according to the state of the adjacent system, although it is an important characteristic of the SoS with high complexity.

This paper proposed the transformation of the SoS-based NCW into an integrated system. For this, we first hypothesized an abstracted combat coupled model that includes the traffic, mobility, and interface atomic models using discrete-event systems specification (DEVS) formalism for the discrete-event simulation. We then estimated the variables of the models in the form of the neural network, which can be updated from the state of the adjacent system, and identified the variable using the machine learning and the data acquired from the SoS-based NCW simulation execution.

The case study shows that the integrated system, as a result of the proposed method, significantly improves the accuracy compared to the existing method and reduces the execution time compared to the SoS-based NCW simulation. We expect that this paper will help in the analysis of various parameters in various domains based on the SoS as well as the military domain.

Glossary

C2:	Command and control
CPU:	Central processing unit
DEVS:	Discrete-event system specification
HLA:	High-level architecture
MANET:	Mobile ad hoc network
M&S:	Modeling and simulation
NCW:	Network-centric warfare
NS3:	Network simulator 3
PDR:	Packet delivery ratio
RAM:	Random-access memory
RMSE:	Root mean square error
RTI:	Run-time infrastructure
RWP:	Random waypoint model
SE:	Standard error
SoS:	System of systems.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean Government (MSIP) (no. 2017-0-00461, Development Platform for User-Level Customizable, General Purpose Discrete-Event Simulation Software).

References

- [1] J. Moffat, "Complexity Theory and Network Centric Warfare," Defense Technical Information Center, 2003.
- [2] B. Xiong, B. Li, R. Fan, Q. Zhou, and W. Li, "Modeling and simulation for effectiveness evaluation of dynamic discrete

- military supply chain networks,” *Complexity*, Art. ID 6052037, 9 pages, 2017.
- [3] J. Cares, *Distributed networked operations: The foundations of network centric warfare*, iUniverse, 2005.
 - [4] C. K. Pang and J. Mathew, “Dynamically reconfigurable command and control structure for network-centric warfare,” *Simulation*, vol. 91, no. 5, pp. 417–431, 2015.
 - [5] D. R. Choffnes and F. E. Bustamante, “An integrated mobility and traffic model for vehicular wireless networks,” in *Proceedings of the VANET - 2nd ACM International Workshop on Vehicular Ad Hoc Networks*, pp. 69–78, deu, September 2005.
 - [6] S. Deller, S. R. Bowling, and G. A. Rabadi, “Applying the Information Age Combat Model: Quantitative Analysis of Network Centric Operations,” *The International C2 Journal*, vol. 3, no. 1, pp. 1–25, 2009.
 - [7] B. G. Kang and T. G. Kim, “Reconfigurable C3 simulation framework: Interoperation between C2 and communication simulators,” in *Proceedings of the 2013 43rd Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC 2013*, pp. 2819–2830, USA, December 2013.
 - [8] A. Tolk, *Engineering principles of combat modeling and distributed simulation*, John Wiley & Sons, 2012.
 - [9] HLA Working Group, IEEE standard for modeling and simulation (M&S) high level architecture (HLA)-framework and rules, IEEE Standard, 2000.
 - [10] U. S. DoD, TENA-the test and training enabling architecture, <https://www.tena-sda.org>.
 - [11] K. H. Lee, J. H. Hong, and T. G. Kim, “System of systems approach to formal modeling of CPS for simulation-based analysis,” *ETRI Journal*, vol. 37, no. 1, pp. 175–185, 2015.
 - [12] K.-M. Seo, K.-P. Park, and B.-J. Lee, “Achieving Data Interoperability of Communication Interfaces for Combat System Engineering,” *IEEE Access*, vol. 5, pp. 17938–17951, 2017.
 - [13] S. H. Choi, K.-M. Seo, and T. G. Kim, “Accelerated simulation of discrete event dynamic systems via a multi-fidelity modeling framework,” *Applied Sciences (Switzerland)*, vol. 7, no. 10, article no. 1056, 2017.
 - [14] K.-M. Seo, W. Hong, and T. G. Kim, “Enhancing model composability and reusability for entity-level combat simulation: A conceptual modeling approach,” *Simulation*, vol. 93, no. 10, pp. 825–840, 2017.
 - [15] C. Choi, K.-M. Seo, and T. G. Kim, “DEXSim: An experimental environment for distributed execution of replicated simulators using a concept of single simulation multiple scenarios,” *Simulation*, vol. 90, no. 4, pp. 355–376, 2014.
 - [16] B. P. Zeigler, T. G. Kim, and H. Praehofer, *Theory of Modeling and Simulation Integrating Discrete Event and Continuous Complex Dynamic Systems*, Academic Press, 2000.
 - [17] T. G. Kim, C. H. Sung, S.-Y. Hong et al., “DEVSIM++ Toolset for Defense Modeling and Simulation and Interoperation,” *The Journal of Defense Modeling and Simulation*, vol. 8, no. 3, pp. 129–142, 2011.
 - [18] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Koppena, “Network simulations with the ns-3 simulator , in SIGCOMM demonstration,” *Network simulations with the ns-3 simulator , in SIGCOMM demonstration*, p. 527, August 2008.
 - [19] J. Pan and R. Jain, *A survey of network simulation tools: Current status and future developments*, Washington University, 2008.
 - [20] M.-W. Yoo, C. Choi, and T. G. Kim, “High-Level Architecture service management for the interoperation of federations,” *Simulation*, vol. 91, no. 6, pp. 566–590, 2015.
 - [21] D. S. Alberts, J. J. Garstka, and F. P. Stein, *Network Centric Warfare: Developing and Leveraging Information Superiority*, Assistant Secretary of Defense (C3I/Command Control Research Program), Washington DC, Wash, USA, 2000.
 - [22] S. K. Sukhpreet Kaur, “An Overview of Mobile Ad hoc Network: Application, Challenges and Comparison of Routing Protocols,” *IOSR Journal of Computer Engineering*, vol. 11, no. 5, pp. 7–11, 2013.
 - [23] M. F. Khan, E. A. Felemban, S. Qaisar, and S. Ali, “Performance analysis on packet delivery ratio and end-to-end delay of different network topologies in wireless sensor networks (WSNs),” in *Proceedings of the 9th IEEE International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2013*, pp. 324–329, China, December 2013.
 - [24] F. Gebali, “Modeling Network Traffic,” in *Analysis of Computer and Communication Networks*, Springer, 2008.
 - [25] F. Bai and A. Helmy, *A survey of mobility models , Wireless Adhoc Networks*, University of Southern, California, 2004.
 - [26] J. M. Choi and Y. B. Ko, “A performance evaluation for ad hoc routing protocols in realistic military scenarios,” in *Proceedings of the in Proceedings of the International Conference on Cellular and Intelligent Communications, CIC, 2004*, October 2004.
 - [27] C. Rajabhushanam and A. Kathirvel, “Survey of Wireless MANET Application in Battlefield Operations,” *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 1, 2011.
 - [28] G. M. Patil, A. Kumar, and A. D. Shaligram, “Performance Analysis and Comparison of MANET Routing Protocols in Selected Traffic Patterns For Scalable Network,” vol. 6, pp. 109–117, 2016.
 - [29] S. Katiyar, R. Gujral, and B. Mallick, “Comparative performance analysis of MANET routing protocols in military operation using NS2,” in *Proceedings of the 1st International Conference on Green Computing and Internet of Things, ICGCIoT 2015*, pp. 603–609, India, October 2015.
 - [30] K. V. Vishwanath and A. Vahdat, “Realistic and responsive network traffic generation,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 111–112, 2006.
 - [31] F. Geyer, S. Schneelee, and G. Carle, “RENETO, a realistic network traffic generator for OMNeT++/INET,” in *Proceedings of the 6th International Conference on Simulation Tools and Techniques, SIMUTools 2013*, pp. 73–81, greece, March 2013.
 - [32] H. S. Bindra, S. K. Maakar, and A. L. Sangal, “Performance evaluation of two reactive routing protocols of MANET using group mobility model,” *International Journal of Computer Science*, vol. 7, no. 3, pp. 38–43, 2010.
 - [33] G. Kioumourtzis, C. Bouras, and A. Gkamas, “Performance evaluation of ad hoc routing protocols for military communications,” *International Journal of Network Management*, vol. 22, no. 3, pp. 216–234, 2012.
 - [34] A. Fongen, M. Gjellerud, and E. Winjum, “A military mobility model for MANET research,” in *Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks, PDCN 2009*, pp. 213–219, aut, February 2009.
 - [35] S. Reidt and S. D. Wolthusen, “An evaluation of cluster head TA distribution mechanisms in tactical MANET environments,” in *Proceedings of the in Proceedings of the International Technical Alliance in Network and Information Science, NIS-ITA, 2007*.
 - [36] X. Lu, Y. C. Chen, I. Leung, Z. Xiong, P. Liò et al., “A novel mobility model from a heterogeneous military MANET trace,” in *in Proceedings of the International Conference on Ad-Hoc*

- Networks and Wireless, ADHOC-NOW*, pp. 463–474, France, September 2008.
- [37] H. Seo, S. H. Kim, and J. S. Ma, “A novel mobility model for the military operations with real traces,” in *Proceedings of the in Proceedings of the International Conference on the Advanced Communication Technology*, pp. 129–133, Korea, February 2010.
- [38] C. M. Seo, B. P. Zeigler, D. H. Kim, and K. Duncan, “Integrating web-based simulation on IT systems with finite probabilistic DEVS,” in *Proceedings of the in Proceedings of the Symposium on Theory of Modeling Simulation: DEVS Integrative MS Symposium*, pp. 173–180, USA, April 2015.
- [39] Y. Kodratoff, *Introduction to machine learning*, Morgan Kaufmann, 2014.
- [40] Z. Xiao, L. Peng, Y. Chen, H. Liu, J. Wang, and Y. Nie, “The Dissolved Oxygen Prediction Method Based on Neural Network,” *Complexity*, vol. 2017, Article ID 4967870, pp. 1–6, 2017.
- [41] T. Gligorijević, Z. Ševarac, B. Milovanović et al., “Follow-up and risk assessment in patients with myocardial infarction using artificial neural networks,” *Complexity*, vol. 2017, Article ID 8953083, 8 pages, 2017.
- [42] G. Neuneck, “The revolution in military affairs: Its driving forces, elements, and complexity,” *Complexity*, vol. 14, no. 1, pp. 50–61, 2008.
- [43] C. E. Maldonado and N. A. Gómez Cruz, “The complexification of engineering,” *Complexity*, vol. 17, no. 4, pp. 8–15, 2012.
- [44] P. Kuosmanen, *Choosing routing protocol for military ad hoc networks based on network structure and dynamics*, Helsinki University of Technology, 2002.
- [45] B. G. Kang, B. S. Kim, and T. G. Kim, “Abstraction on network model under interoperable simulation environment,” in *Proceedings of the 30th European Conference on Modelling and Simulation, ECMS 2016*, pp. 460–466, DEU, June 2016.
- [46] A. H. Dekker, *C4ISR architectures, social network analysis and the FINC methodology: an experiment in military organisational structure*, Defence Science and Technology Organisation, 2002.
- [47] I. C. Moon, K. M. Carley, and T. G. Kim, *Modeling and Simulating Command and Control: For Organizations Under Extreme Situations*, Springer Science & Business Media, 2013.
- [48] I. Porche, R. Isaac, and W. Bradley, “The impact of network performance on warfighter effectiveness,” *RAND ARROYO CENTER SANTA MONICA CA*, 2006.
- [49] I. Porche, L. Jamison, and T. Herbert, *Framework for Measuring the Impact of C4ISR Technologies and Concepts on Warfighter Effectiveness Using High Resolution Simulation*, Rand Arroyo Center Santa Monica CA, 2004.
- [50] N. E. Miner, B. P. Van Leeuwen, K. M. Welch et al., “Evaluating communications system performance effects at a system of systems level,” in *Proceedings of the 2012 IEEE Military Communications Conference, MILCOM 2012*, USA, November 2012.
- [51] S. M. Sanchez and H. Wan, “Better than a petaflop: The power of efficient experimental design,” in *Proceedings of the 2009 Winter Simulation Conference, WSC 2009*, pp. 60–74, USA, December 2009.
- [52] C. K. McCalley, B. J. Woodcroft, S. B. Hodgkins et al., “Methane dynamics regulated by microbial community response to permafrost thaw,” *Nature*, vol. 514, no. 7253, pp. 478–481, 2014.
- [53] J. H. Lee, H. S. Tag, G. T. Kim et al., “Effect of Rheumatoid Factor on Vascular Stiffness in General Population without Joint Symptoms,” *Kosin Medical Journal*, vol. 32, no. 1, p. 25, 2017.

Research Article

Systematic Approach to Optimization for Protection Against Intentional Ultrashort Pulses Based on Multiconductor Modal Filters

Anton O. Belousov  and Talgat R. Gazizov 

Department of Television and Control, Tomsk State University of Control Systems and Radioelectronics, Tomsk, Russia

Correspondence should be addressed to Anton O. Belousov; ant1lafleur@gmail.com

Received 31 December 2017; Revised 9 April 2018; Accepted 18 April 2018; Published 5 June 2018

Academic Editor: Kevin Wong

Copyright © 2018 Anton O. Belousov and Talgat R. Gazizov. 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 protecting radio electronic equipment from ultrashort pulses is of utmost importance nowadays since conductive interference poses the biggest danger to its proper functioning. The article considers the issue of protecting equipment by means of modal filters (MFs) and analyzes the structures of multiconductor microstrip MFs. We present the results of a complex study of the possibility to conduct the optimization (both separate and simultaneous) of a multiconductor MF by different criteria and the formulation of the basic (electrical) optimization criteria for MF. We have formulated the amplitude and time criteria for optimizing an MF (with any number of conductors) in an analytical form and obtained a general multicriteria objective function for optimizing an MF by different criteria. As a result, we have formed a hybrid model consisting of heuristic search and GA. The results demonstrated the topicality of further research in this field.

This study has been carried out in memory of the authors' teacher Ivan Nikolaevich Pustynsky who died on 29.12.2017

1. Introduction

Contemporary radioelectronic equipment has huge functionality but, at the same time, is susceptible to electromagnetic interference. Conducted interference is considered the most harmful one, as it can penetrate into devices directly through conductors [1]. Modern generators of ultrashort pulses have very high capabilities [2]. Such ultrashort pulses are able to penetrate and disturb electronics due to the high power output and short duration. Therefore, it is necessary to improve the protection of electronic equipment against ultrashort pulses.

One of the new protection principles is based on modal filtering—the use of modal distortions (signal changes due to the difference in the mode delay of a multiconductor transmission line (MCTL)) due to the serial modal decomposition of the pulse in segments of coupled lines. A series of researches performed indicate the possibility of creating protection devices based on modal

filtering—modal filters (MF) [3]. They can have such benefits as radiation resistance, low mass, and cost-effectiveness. As a device for protection against pulse disturbances, you can use a strip structure on a widespread foil-coated glass fiber sheet [4]. However, previously, there were studied MFs based only on a pair of coupled lines, while MFs on multiconductor lines are almost unexplored. Thus, the use of MCTL resources in the MF is relevant.

The creation of any new equipment often requires modeling complex systems and solving complex simulation problems. For this aim, there are successfully used optimization methods based on evolutionary algorithms, for example, in power electronics [5] and applied electrodynamics [6]. Unfortunately, a number of trends in developing devices for protection against ultrashort pulses are also increasingly reduced to simulating complex systems.

Indeed, first of all, the number of optimized parameters increases, which in the case of MF is determined by the increase in the number of conductors and dielectrics, as well

as the complexity of their geometry. A significant contribution to the complexity is made by the expansion of the parameter range, as well as the decrease in the step of the optimized parameter, determined by the complex behavior of the characteristics of interest. When optimizing an MF for the parameters, the parameters include not only the conductors and dielectrics of the MF but also the exciting signals. These signals are divided into two large groups: useful and noise. The complexity of the solution can dramatically increase if you optimize not only the parameters, but also the very structure as well. Finally, the convergence of optimization can deteriorate with the growth of the number of simultaneously satisfied criteria (electrical, mass size, and cost) of optimization. Thus, the search for new approaches to reducing computational complexity in the optimization becomes relevant.

The research of separate issues of multiconductor MF optimization has already been carried out [7–11]. However, there is no systematic representation of the recently obtained and new results on solving the complex problem of creating new devices for protecting against ultrashort pulses in a single work. Meanwhile, it is seen important for a wide range of designers of critical electronics. The aim of this paper is to fill this gap. Therefore, it is useful to carefully investigate the propagation of signals in multiconductor MFs and also to formulate the basic criteria for optimizing their parameters. We prefer to start it with the preliminary simulation and optimization of the parameters by heuristic search as it is the simplest and most widely used method of optimization. Further, to validate the results obtained earlier, it is useful to do optimization using the genetic algorithm (GA) as the most popular algorithm for optimization in the problems of electrodynamics and propagation of radio waves [6]. Then, it is expedient to execute the formulation of the basic optimization criteria in an analytical form and proceed from one criterion to multicriteria optimization.

2. Structures and Schematic Diagrams of MFs under Consideration

As the object of investigation, we took multiconductor microstrip lines (MSLs) of lengths $l = 400, 100, \text{ and } 60 \text{ cm}$, consisting of 1–5 conductors. (The length of 400 cm was chosen in the first stage of the investigation to ensure the complete decomposition of the pulse signal in the simulation without considering losses. The lengths of 100 and 60 cm were chosen at the subsequent stages of the investigation for preliminary simulation, which took losses in conductors and dielectrics into account, and for further experimental realization.) The cross sections of these lines are generally shown in Figure 1 where w is the width of conductors, s_i is separations between them, t is the thickness of conductors and h is the thickness of the dielectric, and ϵ_r is the permittivity of the dielectric. The schematic diagrams of these MFs are shown in Figure 2.

First, we constructed geometric models of the MSL cross section. Then, we calculated the matrixes of per-unit-length coefficients of electrostatic \mathbf{C} and electromagnetic \mathbf{L} induction. If it is necessary to take into account the losses, we

calculated the matrixes of the per-unit-length resistances \mathbf{R} (for the losses in the conductors) and conductivities \mathbf{G} (for the losses in the dielectrics). When considering the losses, we used a widely known model [12] of the frequency dependence of the relative permittivity and tangent of the dielectric loss angle of FR-4 material for calculating the \mathbf{G} matrix. The entries of the \mathbf{R} matrix were calculated taking into account the skin effect, the proximity effect, and losses in the ground plane using the method proposed in [13]. Next, we drew a schematic diagram for simulation, set loads and pulse excitation values, computed time response to the excitation in the parameter range, and optimized parameters of a MSL.

It was assumed that a T wave is propagating along the considered lines. The parameters and forms of the signal were calculated in TALGAT software [14], wherein the abovementioned models and steps were implemented. As the exciting pulses, we used a trapezoidal pulse with rise, fall, and flat top durations of 50 ps (the overall duration of 150 ps, to approximate the typical ultrashort pulse in the simulation without considering losses) with an amplitude of 5 V and a digitized signal of the S9-11 oscilloscope (for the maximum possible approximation to the actual exciting signal in the simulation considering losses) with an amplitude of 0.644 V (measured at 50 Ω load) and the durations of rise of 56 ps, fall of 48 ps, and flat top of 4 ps (the overall duration was 108 ps; durations were measured at levels of 0.1–0.9). The waveforms of the exciting pulses are shown in Figure 3.

3. Formulation of Optimization Criteria

Optimization can be carried out by various criteria. The criteria considered below allow us to obtain higher characteristics of a multiconductor MF for protection against ultrashort pulses.

- (1) Minimization of the maximum voltage of an MF output waveform:

$$\max(U(t)) \rightarrow \min. \quad (1)$$

This criterion is the most important one, as signal amplitude at the output of an MF defines its main characteristic: filter attenuation. However, depending on the defined excitation, optimization by this criterion can give various results. It also requires time-consuming computation of the time-domain response that complicates the optimization of complex structures.

- (2) Equalization of decomposition pulse delay differences:

$$\min|t_{i+1} - t_i| \rightarrow \max, \quad i = 1, \dots, Np - 1, \quad (2)$$

where Np is the number of decomposition pulses, t_i is the value of the i th pulse delay.

This criterion is important, as it increases the maximum duration of an initial pulse, which is completely decomposed, and prevents the pulse

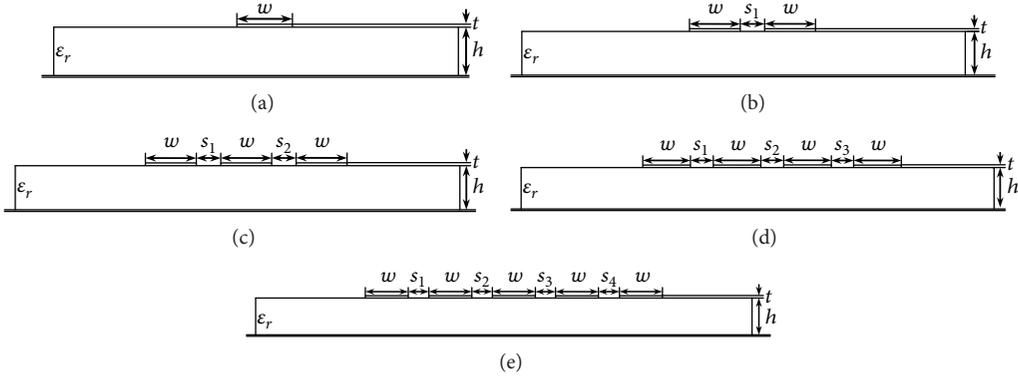


FIGURE 1: Cross sections of one- (a), two- (b), three- (c), four- (d), and five- (e) conductor structures.

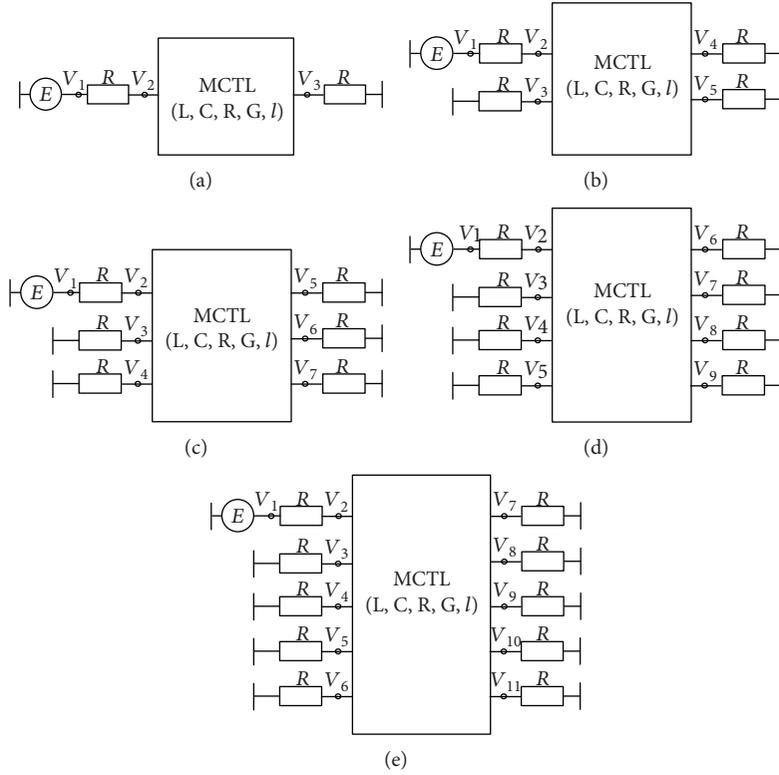


FIGURE 2: Schematic diagrams for one- (a), two- (b), three- (c), four- (d), and five- (e) conductor structures.

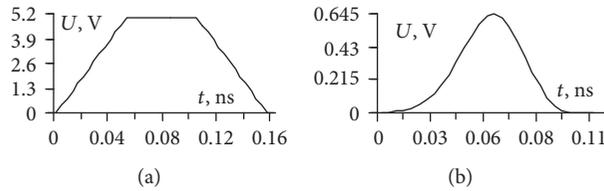


FIGURE 3: EMF waveforms for the overall duration of 150 ps (a) and 108 ps (b).

overlapping, which increases the maximum voltage at the line output.

- (3) Maximization of difference between the maximum and the minimum pulse delays:

$$(t_{\max} - t_{\min}) \rightarrow \max, \quad (3)$$

where t_{\max} and t_{\min} are the maximum and the minimum values of pulse delays, respectively.

This criterion is important, as it additionally increases the maximum duration of an initial pulse, which is completely decomposed. Let us note that in this formulation the criterion can give decomposition pulses in any

TABLE 1: Peak voltage of decomposition pulses, V.

Type of result	$N = 3$ ($s_1 = 400 \mu\text{m}$, $s_2 = 540 \mu\text{m}$)			$N = 4$ ($s_1 = 500 \mu\text{m}$, $s_2 = 675 \mu\text{m}$, and $s_3 = 650 \mu\text{m}$)				$N = 5$ ($s_1 = 367 \mu\text{m}$, $s_2 = 447 \mu\text{m}$, $s_3 = 500 \mu\text{m}$, and $s_4 = 685 \mu\text{m}$)				
Analytics	0.84	0.97	0.67	0.61	0.74	0.64	0.44	0.52	0.66	0.60	0.44	0.23
Simulation	0.84	0.83	0.83	0.70	0.49	0.70	0.60	0.56	0.47	0.39	0.56	0.49

part of a time axis, while the increase of a delay of all pulses can be adverse. In order to minimize it, it is possible to use other formulations:

$$\left(\frac{t_{\min} - l}{c}\right) \rightarrow \min, \quad (4)$$

$$\left(\frac{\varepsilon_{r\max}^{0.5} l}{c - t_{\max}}\right) \rightarrow \min, \quad (5)$$

where l is the length of an MF, c is light velocity in vacuum, and $\varepsilon_{r\max}$ is the maximum value of ε_r of MF dielectrics.

The criterion (4) does make the first pulse delay as short as possible, that is, as determined by light velocity in vacuum. The criterion (5) makes the last pulse delay as long as possible, that is, determined by light velocity in dielectric with the maximum value of ε_r .

4. Accelerated Calculation of Objective Function

Minimization of $\max(U(t))$ demands simulation of a time-domain response. However, it can be computationally expensive, and, therefore, its acceleration is relevant. Meanwhile, while matching the modes of a multiconductor transmission line, amplitudes of decomposition pulses can be computed using analytical expression [6].

$$\mathbf{V} = \mathbf{S}_v \text{diag}(\mathbf{V}_m), \quad (6)$$

where $\mathbf{V}_m = 0.5\mathbf{S}_v^{-1}\mathbf{E}$; \mathbf{S}_v is a matrix of $N \times N$ size that contains eigenvectors of LC matrix; \mathbf{E} is a vector of $N \times 1$ size that consists of the values of voltage source amplitudes, where N is a number of conductors.

As for maximization of $\min|t_{i+1} - t_i|$ and maximization of $(t_{\max} - t_{\min})$, generally, they also require calculation of a time-domain response, as we need the values of the moments when decomposition pulses arrive at the end of the line. However, practical experience in simulating an MF shows that for the matched MF and weak coupling between the conductors, the maximum amplitude of pulses is defined by amplitudes of the first traveling wave pulses (subjected to one reflection at most). The moments of their arrival to the end of MF are defined by values of per-unit-length delay of modes. In such case, for optimization, there is no need to calculate a time-domain response, and it is enough to calculate only per-unit-length delay that considerably reduces computational costs. Optimization criteria are slightly changed. So, (2) takes a form of

$$\min|\tau_{i+1} - \tau_i| \rightarrow \max, \quad i = 1, \dots, N-1, \quad (7)$$

where N is the number of conductors, τ_i is the value of per-unit-length delay of the i th pulse. From (3), we obtain

$$(\tau_{\max} - \tau_{\min}) \rightarrow \max, \quad (8)$$

where τ_{\max} and τ_{\min} are the maximum and the minimum values of per-unit-length delay. From (4) and (5), we obtain

$$\left(\frac{\tau_{\min} - l}{c}\right) \rightarrow \min, \quad (9)$$

$$\left(\frac{\varepsilon_{r\max}^{0.5}}{c - \tau_{\max}}\right) \rightarrow \min.$$

Thus, the presented analytical expressions can considerably accelerate calculation of objective function.

5. MF Optimization by Heuristic Search

5.1. Optimization by Criterion of Minimizing the Maximum Voltage at the MF Output. To begin with, we can optimize the parameters by using heuristic search for MF with $N = 2, 3, 4, 5$ by the criterion of minimizing the maximum amplitude of the output signal. Losses in conductors and dielectrics were not taken into account. As for the exciting pulses, we used signal shown in Figure 3(a).

For $N = 2$, the absolute value of difference of modal per-unit-length delays is 0.34 ns/m at $s = 500 \mu\text{m}$, $w = 290 \mu\text{m}$, $t = 105 \mu\text{m}$, $h = 190 \mu\text{m}$ (standard material), and $\varepsilon_r = 5$. The value of w was optimized, in order to assure characteristic impedance of a single line to be equal to 50Ω , and it was kept unchanged thereafter, just as the values of t , h , and ε_r . For $N = 2$, we took the typical values of the parameters s , t , h , and ε_r for the technology of printed circuit boards. The length of the line was 400 cm . Each pair of neighboring conductors has its own gap between the conductors (s_i), and these values were optimized in the range of $1\text{--}1000 \mu\text{m}$ to equalize delay differences between decomposition pulses at the output of an active conductor of the MSL.

Generally, 3, 4, and 5 modes are propagating along the line with $N = 3, 4, 5$ correspondingly, and each mode has its own characteristics. The optimized values of s_i for $N = 3, 4, 5$, as well as the values of the amplitudes of the decomposition pulses, calculated analytically and by simulation with the EMF source with the amplitude of 5 V connected between conductor 1 (active) and the reference conductor, are presented in Table 1. The voltage waveforms at the input and output of the lines are shown in Figure 4. One can see 3, 4, and 5 pulses with maximum amplitudes, which are by 3, 3.6, and 4.5 times, respectively, less than the signal level at the beginning of the line. Thus, adding more conductors with identical parameters to the two-conductor structure, with identical parameters, significantly (almost by 5 times) decreases the amplitude at the line output.

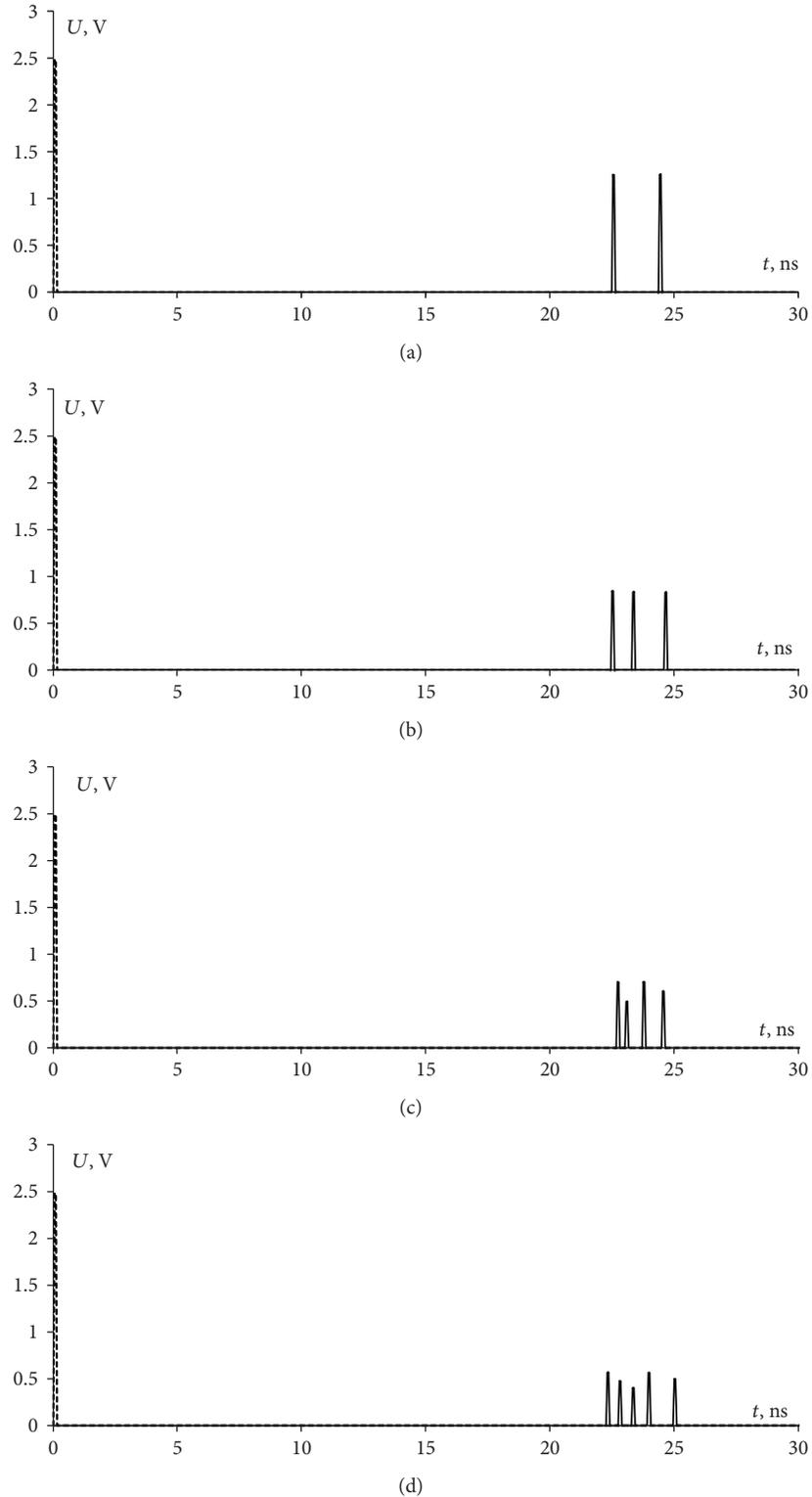


FIGURE 4: Voltage waveforms at the input (---) and the output (—) of conductor 1 for two- (a), three- (b), four- (c), and five- (d) conductor MFs with parameters, obtained by heuristic search optimization.

5.2. *Optimization by Criterion of Equalizing the Time Intervals between the Decomposition Pulses.* Figure 4 shows a decrease in the pulse amplitude at the MF output with increasing N . However, the difference of the maximum and

minimum delays has remained unaddressed that has led to an uncontrollable variation of this value in different structures and, as a result, has limited the maximum duration of the initial pulse which will be decomposed completely.

TABLE 2: Values of s_i and per-unit-length delays after optimization using a heuristic search.

Parameter	$N = 3$ ($s_1 = 170 \mu\text{m}$, $s_2 = 540 \mu\text{m}$)			$N = 4$ ($s_1 = 70 \mu\text{m}$, $s_2 = 335 \mu\text{m}$, and $s_3 = 250 \mu\text{m}$)			$N = 5$ ($s_1 = 15 \mu\text{m}$, $s_2 = 32 \mu\text{m}$, $s_3 = 290 \mu\text{m}$, and $s_4 = 200 \mu\text{m}$)					
τ_i , ns/m	5.37	5.81	6.25	5.02	5.49	5.96	6.43	4.2	4.8	5.4	6	6.6

Obviously, when applying the pulse, with a total duration of more than 200 ps, with the same line parameters, we will observe the overlapping of the pulses, and as a consequence, an increase in the amplitude of the signal at the end of the line. Thus, it is also relevant to make optimization by another criterion—equalization of decomposition pulse delay differences.

The parameters s_i (with the previous values of the remaining parameters) were optimized by the specified criterion using heuristic search for MFs with $N = 3, 4, 5$. Losses in conductors and dielectrics were not taken into account.

The optimized values of s_i , as well as the per-unit-length modal delays, are given in Table 2. All values of s_i are significantly different from the values obtained by the criterion of amplitude decrease, because equalization of delay differences between decomposition pulses is achieved by means of increasing the coupling between the external (active) and central conductors.

Table 3 gives the values of the maximum amplitudes of decomposition pulses at the end of the active conductor for $N = 3, 4, 5$ for different optimization criteria computed with (6) analytically ($\max(\mathbf{V}_A)$) and according to the time response ($\max(\mathbf{V}_R)$), as well as the minimum values of differences of modal per-unit-length delays.

It is clear from Table 3 that when equalization of absolute values of differences of modal per-unit-length delays ($\Delta\tau_i$) is taken as an optimization criterion, the value of $\min(\Delta\tau_i)$ increased by 1.4 times simultaneously with the increase of N , and, in comparison with the optimization that took $\max(\mathbf{V})$ as an optimization criterion, it increased by 2 times for $N = 3$ and by 5 times for $N = 4$ and $N = 5$.

However, improvement of one parameter leads to degradation of another. As shown in Table 3, the maximum amplitudes of decomposition pulses are 2.2, 2.3, and 3.9 times less than the pulse amplitude at the near end of the line, which is less than the corresponding values (3, 3.6, and 4.5) for optimization that take amplitudes of an output signal as an optimization criterion.

There is a tendency for increasing the difference between the values of the maximum amplitudes of decomposition pulses, calculated analytically and according to the response. This tendency appears when the number of N starts to grow and is the strongest for $N = 5$. It can be explained by incomplete matching of a transmission line. However, the very fact that the maximum amplitudes of decomposition pulses have been accurately computed using (6) is rather important; as in the future, it provides fast optimization by using an analytical formula, that is, it will not be necessary to compute time response.

Apparently, different results can be obtained for different aims. For example, in optimization by equalizing decomposition pulse delay differences, central conductors are seen to be

TABLE 3: Results of MF optimization for $N = 3, 4, 5$ separately by two criteria.

Criterion	$\max(U(t)) \rightarrow \min$			$\min \tau_{i+1} - \tau_i \rightarrow \max$		
	3	4	5	3	4	5
$\min(\Delta\tau_i)$, ns/m	0.21	0.09	0.12	0.44	0.47	0.6
$\max(\mathbf{V}_A)$, V	0.97	0.74	0.66	1.16	1.15	0.9
$\max(\mathbf{V}_R)$, V	0.84	0.70	0.56	1.14	1.08	0.65

closer to external ones, but this optimization does not give the minimum level of ultrashort pulse at the line output.

As we have already mentioned above, unaccounted delay differences between pulses can cause incomplete decomposition of a pulse and, at the same time, the increase in its duration, and, consequently, it can cause overlapping of pulses that leads to the increase of pulse amplitudes at the end of a line. But if the overlapped pulses together do not exceed the maximum amplitude of the rest pulses, then this overlapping can be considered not crucial.

5.3. Optimization by Criterion of Maximizing the Difference between the Maximum and the Minimum Pulse Delays. Note that previously the difference of the maximum and minimum delays has remained unaddressed, which has led to an uncontrollable variation of its value in different structures and, as a result, has limited the maximum duration of an initial pulse that will be decomposed completely. Thus, it is also relevant to conduct optimization by another criterion—maximization of the difference between the maximum and the minimum pulse delays.

The parameters s_i (with the previous values of the remaining parameters) were optimized by the specified criterion using heuristic search for MF for $N = 3, 4, 5$. The losses in conductors and dielectrics were not taken into account. The MFs were simulated with $w = 290 \mu\text{m}$, $t = 105 \mu\text{m}$, $h = 190 \mu\text{m}$, and $\epsilon_r = 5$. As before, the parameters s_i were optimized. For $N = 3$, we considered two values of the lower limit of range s : 50 and $1 \mu\text{m}$. The first was taken, for practical reasons, as the minimum for printed circuit boards. The second was taken to assess limit values of the characteristics of interest.

Table 4 shows the optimization results for $N = 3, 4, 5$ separately by three different criteria: minimization of $\max(U(t))$, maximization of $\min|t_{i+1} - t_i|$, and maximization of $(t_{\max} - t_{\min})$.

The maximum amplitudes of pulses at the output of MF are calculated by (6) and represented in the last line of Table 4. As you can see, they are very close to the amplitudes obtained from a response. (The exception is the last case of a very close coupling, for which an error of expression (6) is too high, as it does not consider reflection of modes.) Thus,

TABLE 4: Results of MF optimization for $N = 3, 4, 5$ using different criteria.

Criterion	$\max(U(t)) \rightarrow \min$			$\min t_{i+1}-t_i \rightarrow \max$			$(t_{\max}-t_{\min}) \rightarrow \max$			
	3	4	5	3	4	5	3	4	5	
$s_p, \mu\text{m}$	400, 540	500, 675, 650	367, 447, 500, 685	170, 540	70, 335, 250	15, 32, 290, 200	50, 50	1, 10	50, 50, 50	50, 50, 50, 50
$\tau_{\max}-\tau_{\min}$, ns/m	0.534	0.45	0.67	0.88	1.4	2.39	1.78	3	1.98	2.11
$t_{\max}-t_{\min}$, ns	2.136	1.8	2.68	3.52	5.6	9.56	7.12	12	7.92	8.44
$\max(\mathbf{V}_R)$, V	0.842	0.70	0.56	1.14	1.08	0.65	1.22	0.8	1.05	0.87
$\max(\mathbf{V}_A)$, V	0.838	0.70	0.56	1.16	1.16	0.68	1.25	1.27	1.06	0.88

TABLE 5: Results of the GA optimization of s_1 (μm) by minimizing the amplitude of the signal ($\max(U(t))$ (V)) at the three-conductor MF output.

N	Number of individuals, number of generations							
	3, 10		10, 10		10, 30		10, 100	
	s_1	$\max(U(t))$	s_1	$\max(U(t))$	s_1	$\max(U(t))$	s_1	$\max(U(t))$
1	366	0.0373535	333	0.0364478	323	0.0366652	329	0.0364521
2	264	0.0387915	332	0.0364339	325	0.0365686	330	0.0364266
3	273	0.0384555	328	0.0364779	327	0.0364996	329	0.0364521
4	345	0.0364985	325	0.0365686	327	0.0364996	330	0.0364266
5	319	0.0368421	326	0.0365231	324	0.0366183	330	0.0364266

these examples confirm the possibility of rapid optimization by an analytical formula, that is, without time-consuming response calculations.

For $s_i \geq 50 \mu\text{m}$, the optimized values of s_i are in the lower limit of the range ($s_1 = s_2 = 50 \mu\text{m}$), and for $s_i \geq 1 \mu\text{m}$, they are near it ($s_1 = 1 \mu\text{m}$, $s_2 = 10 \mu\text{m}$). They considerably differ from the values obtained earlier, which are clear from the first line of Table 4. For $N = 3$ and $s_1 = s_2 = 50 \mu\text{m}$, the value ($\tau_{\max} - \tau_{\min}$) has increased twice in comparison with the previous criterion, and for $s_1 = 1 \mu\text{m}$, $s_2 = 10 \mu\text{m}$, it is 1.7 times more additionally, and it was revealed that it is considerably influenced only by s_1 . The value ($t_{\max} - t_{\min}$) has also increased by 1.7 times. For $N = 4$ and $s_i = 50 \mu\text{m}$, the value ($\tau_{\max} - \tau_{\min}$) has increased by 1.4 times in comparison with the previous criterion, and for $N = 5$, it has decreased by 1.1 times, and it was revealed that it is considerably influenced only by s_1 , which value in the previous criterion was $15 \mu\text{m}$ for $N = 5$.

As you can see, the improvement of one parameter (the increase of delay difference) can worsen another (the increase of the maximum pulse amplitude). For example, when changing over from minimization of $\max(U(t))$ to maximization of $(t_{\max} - t_{\min})$, the value of $\max(U(t))$ increases by 1.45 times for $N = 3$, by 1.5 times for $N = 4$, and by 1.55 times for $N = 5$. However for $N = 3$, when s_i is reduced, the value of $\max(U(t))$ decreases to 0.8 V, yielding the best result.

6. Hybrid Model of MF Optimization by Heuristic Search and Genetic Algorithm

Heuristic search of parameters was used in Section 5. This method was sufficient to achieve one or another criterion, but does not guarantee the best results. Therefore, it is better to use optimization techniques to improve earlier results. It is

also indicative to compare the obtained results with the results of heuristic search.

6.1. Optimization by Heuristic Search. In Section 5.1, the optimization of the three-conductor MF by heuristic search has been carried out; however, in this Section, in order to be closer to real conditions, we used a real initial pulse (Figure 3(b)) with losses in the conductors and dielectrics being taken into account. Thus, to obtain consistent comparison results, it is necessary to perform the optimization by heuristic search once again.

The MF was optimized for $w = 1000 \mu\text{m}$, $t = 18 \mu\text{m}$, $h = 500 \mu\text{m}$, $\epsilon_r = 4.5$, and $l = 60 \text{ cm}$. The value of w was preliminarily optimized in order to assure 50Ω characteristic impedance of a single line, and it was kept unchanged, as well as the values of t , h , and ϵ_r , taken in accordance with the parameters of a real printed circuit board. The values of conductor separations were optimized by minimizing the maximum voltage at the MF output. As a result of heuristic search, we obtained the values $s_1 = 200 \mu\text{m}$, $s_2 = 685 \mu\text{m}$. The voltage waveforms at the input and output of the three-conductor MF with the parameters resulting from the heuristic search are presented in Figure 3. The amplitude of the output voltage was 0.040925 V.

6.2. Optimization by Genetic Algorithm. We used a simple GA implemented in the TALGAT system. The GA was run with typical values (which were not changed further) of the mutation (0.1) and the crossover (0.5) coefficients with a different number of individuals and generations (Table 5). Optimization of s_1 and s_2 was performed in the range of $\pm 200 \mu\text{m}$ from the values obtained by heuristic search.

It follows from Section 5.1 that the lowest possible level of amplitude in a three-conductor MF can be achieved by

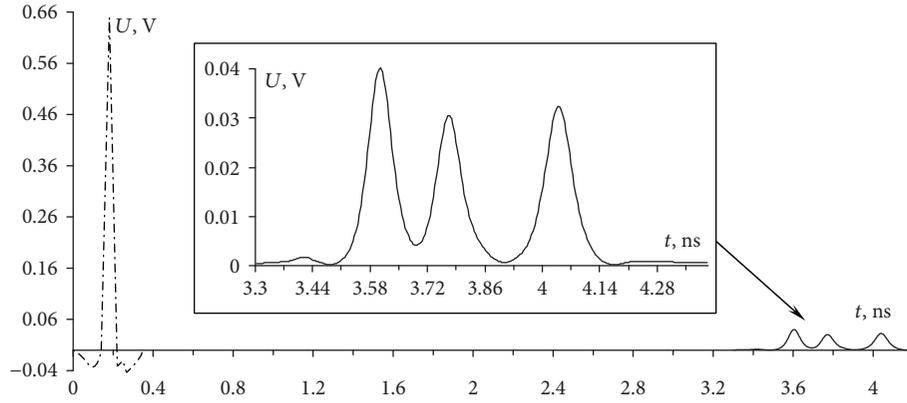


FIGURE 5: Voltage waveforms at the input (---) and output (—) (with enlarged fragment of the signal at the output) of a three-conductor microstrip line MF with the parameters obtained as a result of heuristic search.

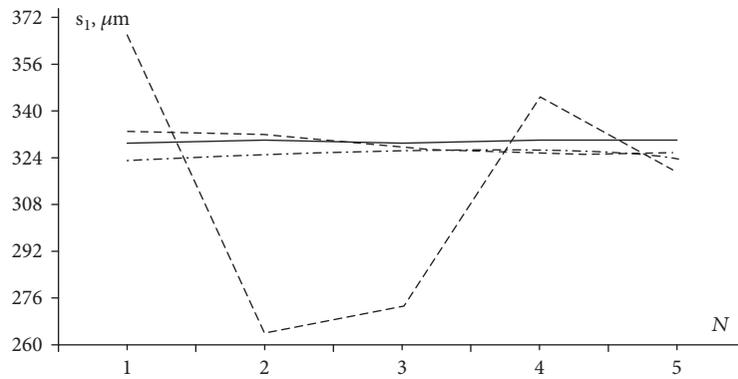


FIGURE 6: The results of GA optimization of s_1 value for the number of individuals and generations: 3 and 10 (---); 10 and 10 (···); 10 and 30 (-·-); and 10 and 100 (—).

equalizing the amplitudes of the decomposition pulses. Meanwhile, Figure 5 shows the possibility to perform further minimization of the pulse amplitudes by their alignment. According to what is stated above, at first, GA was used to optimize s_1 value only, because the heuristic search revealed that it is the s_1 that strongly influences the output waveform. The results for the n th run of the GA are presented in Table 5, and the graphs for s_1 are shown in Figure 6.

When the number of individuals and generations was 3 and 10, respectively (30 calculations), the difference between the extreme values of s_1 was 38.6%. However, for the number of individuals and generations equal to 10 and 10, respectively (100 computations), this difference was 2.5%, for 10 and 30 (300 calculations)—1.23%, and for 10 and 100 (1000 calculations)—0.3%. As a result, when $s_1 = 330 \mu\text{m}$, we obtained the minimum amplitude of 0.03642 V (Table 5), which is 12.4% less than that in the heuristic search.

It follows from what is presented in Table 5 that with the increase in the number of individuals and generations of a GA, the fluctuation of s_1 value decreases (from 38.6% to 0.3%). As a result, for $s_1 = 330 \mu\text{m}$, we obtained the minimum amplitude, which is 12.4% less than with that in the heuristic search. The voltage waveforms at the input and output of the three-conductor MF with parameters after optimization are presented in Figure 7. It is seen that the amplitudes of the first

and second pulses are nearly aligned and the third amplitude is smaller. This could mean the possibility of further reducing the amplitude of the output signal by means of simultaneous optimization of separations s_1 and s_2 . The results of such optimization for 10 individuals with a change in the number of generations are given in Table 6, and the graphs for s_1 and s_2 are shown in Figure 8. It is seen that with the increase in the number of generations, the fluctuation of the s_1 and s_2 values decreases (from 19.3% and 18.5% to 0.6% and 0.7%, resp.).

As a result, when $s_1 = 330 \mu\text{m}$ and $s_2 = 675 \mu\text{m}$, we obtain the maximum possible minimization of the amplitude at the MF output (in this case, the alignment of the first and second pulses) and it is equal to 0.036195 V (Table 6). It is worth noting that the optimized parameter s_1 has a value that is equal to the previous one (Table 5) and the change in the parameter s_2 by 1.5% (from 685 to 675 μm) helped to align the first and second pulses, thus providing the lowest signal amplitude at the MF output. The voltage waveforms at the input and output of a three-conductor MF with the parameters obtained through optimization using GA are shown in Figure 9.

By optimizing two parameters using GA, we achieved the maximum attenuation of the output signal, obtaining the level of 0.036195 V, which is 0.6% less than the one obtained

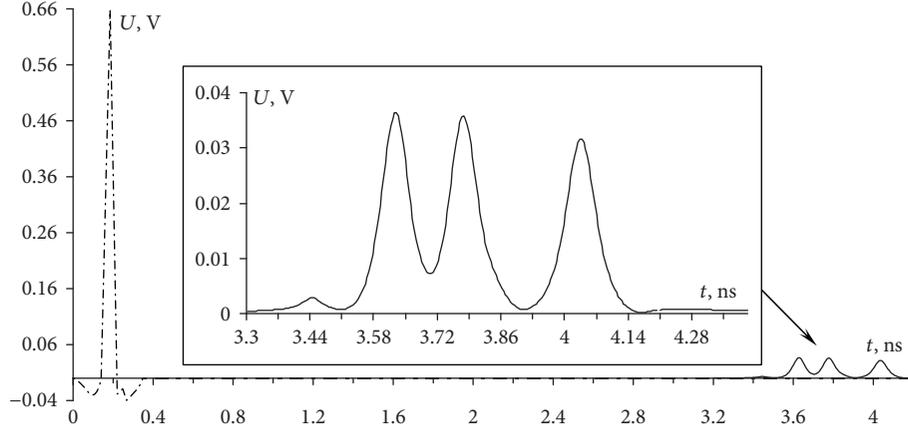


FIGURE 7: Voltage waveforms at the input (---) and output (—) (with enlarged fragment of the signal at the output) of a three-conductor microstrip line MF after optimization of parameter s_1 by GA.

TABLE 6: Results of the GA optimization of s_1 and s_2 (μm) by minimizing the amplitude of the signal ($\max(U(t))$ (V)) at the three-conductor MF output.

N	Number of generations								
	10			30			100		
	s_1	s_2	$\max(U(t))$	s_1	s_2	$\max(U(t))$	s_1	s_2	$\max(U(t))$
1	389	788	0.037411	350	702	0.036216	330	678	0.036198
2	328	676	0.036188	329	675	0.036196	329	675	0.036196
3	335	680	0.036231	334	679	0.036224	330	675	0.036195
4	359	699	0.036470	339	685	0.036213	331	676	0.036203
5	326	665	0.036367	328	669	0.036318	331	673	0.036311

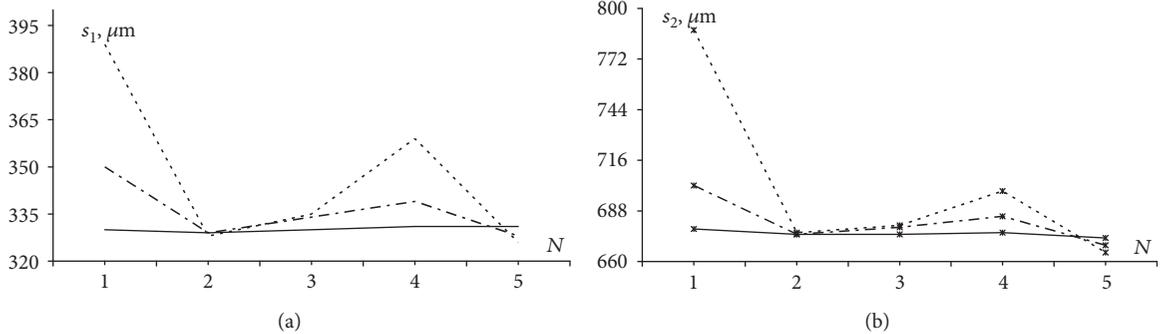


FIGURE 8: The results of GA optimization of s_1 (a) and s_2 (b) values for 10 individuals and 10 (···), 30 (---), and 100 (—) generations.

by optimization of a single parameter and 13% less than the level obtained by heuristic search. Thus, it is reasonable to use a hybrid model of optimization that should include heuristic search and the use of GA.

6.3. Multicriteria Optimization Using GA. In the previous Sections, we used only one criterion for optimization. As it was noted in Section 5.2, depending on the required aim, one can obtain certain results, that is, when one criterion is reached, another deteriorates. However, it is possible to achieve an optimum, in which all the necessary optimization

criteria will be observed as much as required. Thus, it is expedient to formulate a general objective function for the optimization by several criteria and to formulate basic optimization criteria.

6.3.1. General Formulation of the Multicriteria Objective Function. The formulation of a multicriteria objective function (F) implies combining separate criteria to a single problem of minimization or maximization:

$$F \rightarrow \min \text{ or } F \rightarrow \max. \quad (10)$$

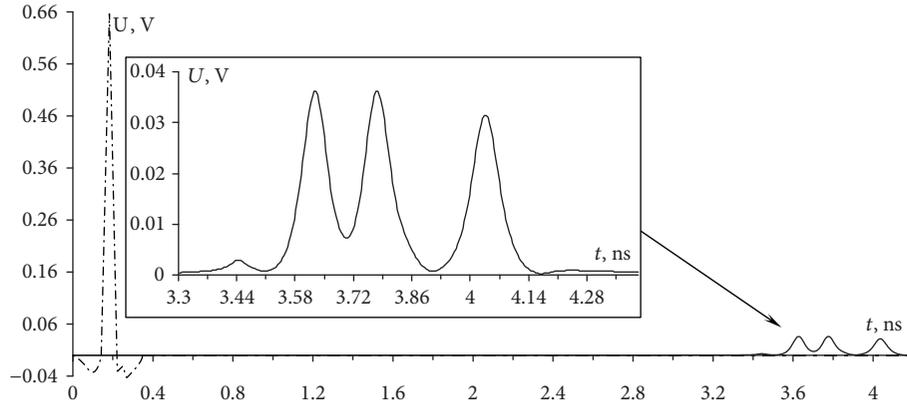


FIGURE 9: Voltage waveforms at the input (---) and output (—) (with enlarged fragment of the signal at the output) of a three-conductor microstrip line MF after optimization of parameters s_1 and s_2 by GA.

For brevity, we will consider the minimization. For example, one can minimize the sum or the maximum of weighted and normalized absolute values of the objective functions that formulate separate criteria:

$$F = \sum_i |F_i| \text{ or } F = \max |F_i|, \quad (11)$$

where

$$F_i = M_i \frac{f_i}{K_i}, \quad (12)$$

where f_i is the objective function, K_i is the normalization constant, M_i is the weighting coefficient of the i th criterion, and $i = 1, 2, \dots, N_C$, where N_C is the number of optimization criteria.

The normalization coefficients K_i are chosen to be equal to the maximum possible value of the i th objective function so that the value of f_i/K_i becomes dimensionless and takes values from 0 to 1 during optimization. Moreover, K_i must guarantee nonnegative values of F_i . The significance of the i th criterion is given by the weighting coefficients M_i . If the criteria are of equal value to the user, then these coefficients are the same and can be given as

$$M_i = \frac{1}{N_C}. \quad (13)$$

Optimization can be performed according to various criteria. Amplitude and time criteria are relevant to the electrical characteristics of multiconductor MFs. They are discussed in detail in the following sections.

6.3.2. Amplitude Criteria. The most important criteria for optimization of an MF are amplitude ones. They can be considered in the time and frequency domains. It is useful to analyze voltage waveforms $U(t)$ at the MF output to provide protection against the exciting ultrashort pulse of electromotive force $E(t)$. Therefore, let us consider the amplitude criteria in the time domain. On the basis of $U(t)$, five norms used to evaluate the effectiveness of ultrashort

pulses impact on different (according to specificity of the response to the impact) equipment [2]. Using these norms, we can formulate expressions for f_i and K_i .

- (1) For the circuit upset, as well as electric breakdown or arc-over effects, the maximum magnitude of the $U(t)$ is important:

$$\begin{aligned} f_1 &= \max |U(t)|, \\ K_1 &= \max |E(t)|. \end{aligned} \quad (14)$$

- (2) For component arcing, as well as the circuit upset, the maximum magnitude of the $U(t)$ change rate is important:

$$\begin{aligned} f_2 &= \max \left| \frac{\partial U(t)}{\partial t} \right|, \\ K_2 &= \max \left| \frac{\partial E(t)}{\partial t} \right|. \end{aligned} \quad (15)$$

- (3) For dielectric puncture, the maximum magnitude of the integral of the $U(t)$ is important:

$$\begin{aligned} f_3 &= \max \left| \int_0^t U(t) dt \right|, \\ K_3 &= \max \left| \int_0^t E(t) dt \right|. \end{aligned} \quad (16)$$

- (4) For equipment damage, the integral of the $U(t)$ magnitude is important:

$$\begin{aligned} f_4 &= \int_0^{\infty} |U(t)| dt, \\ K_4 &= \int_0^{\infty} |E(t)| dt. \end{aligned} \quad (17)$$

TABLE 7: Optimized parameters of MF for $N = 3$, their ranges, and values after optimization with different weighing coefficients.

Parameter	Range	Value		
		$M_1 = M_2 = M_3 = M_4 = 0.25$	$M_1 = M_2 = M_3 = 0.3, M_4 = 0.1$	$M_1 = 0.1, M_2 = M_3 = M_4 = 0.3$
$t, \mu\text{m}$	10–200	174	176	200
$h, \mu\text{m}$	200–2000	995	200	201
$s_1, \mu\text{m}$	1–1000	10	4	3
$s_2, \mu\text{m}$	1–1000	115	48	44

- (5) For component burnout, the square root of the integral of the square of the $U(t)$ magnitude is important:

$$f_5 = \left\{ \int_0^{\infty} |U(t)|^2 dt \right\}^{1/2}, \quad (18)$$

$$K_5 = \left\{ \int_0^{\infty} |E(t)|^2 dt \right\}^{1/2}.$$

6.3.3. Time Criteria. Time criteria are important for preventing pulses that increase the maximum voltage at the MF output with increasing duration of the exciting ultrashort pulse. In contrast to amplitude ones, time criteria may not require costly computation of the response, since it is enough to calculate only the per-unit-length delays. We consider three types of time criteria.

The minimum-time criterion and the maximum-time criterion are associated with the expansion of the pulse time range at the MF output. The time interval criterion is related to the alignment of time intervals. Note that, in time criteria, the values of a per-unit-length delay are sorted out in ascending order.

The minimum-time criterion makes a per-unit-length delay of the first pulse (τ_{\min}) as short as possible, that is, as determined by the light velocity in vacuum:

$$f_6 = \tau_{\min} - \frac{1}{c}, \quad (19)$$

$$K_6 = \frac{\sqrt{\epsilon_r \max} - 1}{c}.$$

The maximum-time criterion makes a per-unit-length delay of the last pulse (τ_{\max}) as long as possible, that is, as determined by the light velocity in dielectric with the maximum value of the relative dielectric permittivity ($\epsilon_r \max$):

$$f_7 = \frac{\sqrt{\epsilon_r \max}}{c} - \tau_{\max}, \quad (20)$$

$$K_7 = \frac{\sqrt{\epsilon_r \max} - 1}{c}.$$

To expand the time range in both directions, these criteria must be used together. They are applicable to an MF with any value of N .

The time interval criterion is important when $N > 2$. It is used to equalize time intervals between the pulses at the MF output. It allows increasing the duration of the exciting ultrashort pulse, which will be decomposed at the MF output completely. For the values of per-unit-length delays which

are sorted out in an ascending order, and based on the deviation of the current values of the per-unit-length delays of the intermediate modes from the values according to the uniform time intervals between the pulses, we obtain

$$f_8 = \max |\tau_i - (\tau_{\min} + (i-1) \cdot \Delta)|, \quad i = 2, \dots, N-1, \quad (21)$$

$$K_8 = \frac{\sqrt{\epsilon_r \max} - 1}{c},$$

where

$$\Delta = \frac{\tau_{\max} - \tau_{\min}}{N-1}, \quad (22)$$

where τ_i is the value of a per-unit-length delay of the i th pulse.

6.4. Multicriteria GA Optimization of MF with $N = 3$. To test the theory, a three-conductor microstrip MF was optimized by GA. We used the multicriteria objective function that combines one amplitude (14) and three (19), (20), and (21) time criteria with equal weighting coefficients for $N = 3$:

$$F = M_1 \cdot \left(\frac{\max(U(t))}{\max(E(t))} \right) + M_2 \cdot \left(\frac{\tau_1 - (1/c)}{(\sqrt{\epsilon_r \max} - 1)/c} \right) + M_3 \cdot \left(\frac{(\sqrt{\epsilon_r \max}/c) - \tau_3}{(\sqrt{\epsilon_r \max} - 1)/c} \right) + M_4 \cdot \left(\frac{|2\tau_2 - \tau_1 - \tau_3|}{(\sqrt{\epsilon_r \max} - 1)/c} \right). \quad (23)$$

The GA parameters were chosen (and further remained unchanged) as follows: number of individuals was 50; number of generations was 100.

The MF was optimized for the following parameters: $w = 1000 \mu\text{m}$, $\epsilon_r = 5$, $l = 60 \text{ cm}$, and $R = 50 \Omega$. The value of w was optimized in order to provide 50Ω characteristic impedance of a single line, and it was unchanged, as well as the value of ϵ_r . The values of t , h , and s_1 , s_2 were optimized by the multicriteria objective function (23). The optimized parameters, their ranges, and values after optimization with different weighing coefficients are given in Table 7. For $M_1 = M_2 = M_3 = M_4 = 0.25$, the amplitude was minimized to 0.03066 V , the equalized values of $\Delta\tau_i$ were 1.24317 and 1.24372 ns/m , so their difference was minimized to 0.00055 ns/m , and the value $(\tau_{\max} - \tau_{\min})$ was maximized to 2.48689 ns/m . The voltage waveforms at the input and output of the MF with the parameters after the GA optimization are presented in Figure 10. Thus, the attenuation

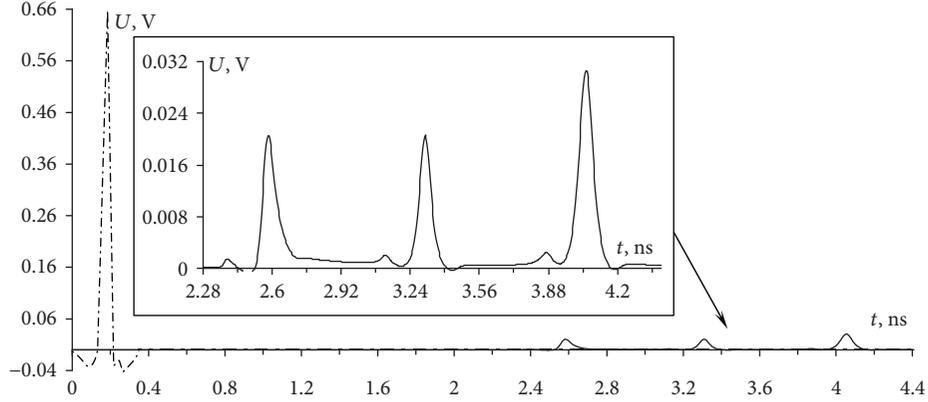


FIGURE 10: Voltage waveforms at the input (---) and output (—) (with enlarged fragment of the signal at the output) of a three-conductor microstrip line MF after four-criterion GA optimization.

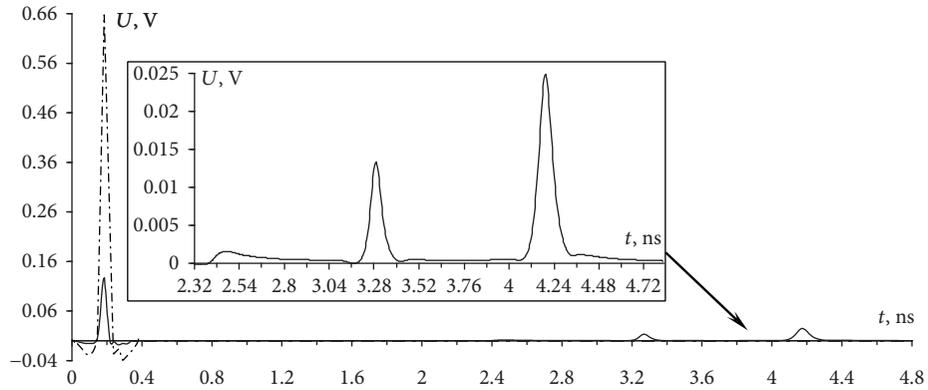


FIGURE 11: Voltage waveforms at the input (---) and output (—) (with enlarged fragment of the signal at the output) of a three-conductor microstrip line MF after the four-criterion GA optimization with the objective function (23), when time interval criterion is less important.

factor of 21.4 is possible for ultrashort pulse with the duration of less than 0.6 ns.

It is also useful to perform multicriteria optimization using 4 criteria (23), but with different weighing coefficients. Suppose that for a designer the amplitude and the maximum-time and minimum-time criteria are essential, while the time interval criterion is less important. Table 7 shows the values of the parameters after optimization for $M_1 = M_2 = M_3 = 0.3$ and $M_4 = 0.1$. In this case, the amplitude was minimized to 0.0247953 V, the equalized values of $\Delta\tau_i$ were 1.50777 and 1.38147 ns/m, so their difference was minimized to 0.1263 ns/m, and the value $(\tau_{\max} - \tau_{\min})$ was maximized to 2.88924 ns/m. The voltage waveforms at the MF input and output with parameters after the GA optimization are presented in Figure 11. For this once suppose that for a designer the maximum-time, minimum-time and time-interval criteria are essential, while the amplitude criterion is less important. Then we can define $M_1 = 0.1$ and $M_2 = M_3 = M_4 = 0.3$. Table 7 shows the values of the parameters after this optimization. In this case, the amplitude was minimized to 0.0249575 V, the equalized values of $\Delta\tau_i$ were 1.501 and 1.505 ns/m, so their difference was minimized to 0.004 ns/m, and the value $(\tau_{\max} - \tau_{\min})$ was

maximized to 3.00546 ns/m. The voltage waveforms at the input and output of the MF with parameters after the GA optimization are presented in Figure 12.

6.5. Multicriteria Optimization of MF with $N = 4$ with Respect to the Criterion of Matching. Previously, the matching of tract was neglected for $N > 3$. In addition, multicriteria optimization of a microstrip MF for $N > 3$ by GA has not been performed before. However, it is relevant, since the increase in the number of conductors, in general, improves the characteristics of the MF. Thus, multicriteria optimization of a four-conductor microstrip MF with respect to the matching criterion is useful.

6.5.1. Formulation of the Matching Criterion. One of the important criteria for optimizing an MF is the matching criterion. This criterion is important in order to minimize the reflection of useful high-frequency signals from the MF.

The condition for matching the two coupled lines by resistances R at their ends is defined as the geometric mean value of the impedances of the even and odd modes [15]:

$$R = \sqrt{Z_e Z_o}. \quad (24)$$

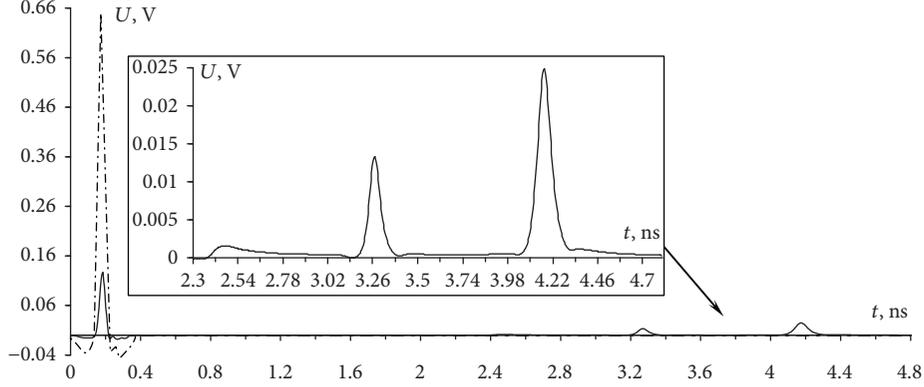


FIGURE 12: Voltage waveforms at the input (---) and output (—) (with enlarged fragment of the signal at the output) of a three-conductor microstrip line MF after the four-criterion GA optimization with the objective function (23), when amplitude criterion is less important.

In multiconductor transmission lines, the number of propagating modes is equal to the number of conductors (N). From the eigenvalues of the impedance matrix of the line, it is possible to determine the mode impedances. The subsequent stages of determining the matching condition are apparently possible on the basis of the theory of multiconductor transmission lines, but are not clear for authors yet. Meanwhile, by analogy with a matched single transmission line, to match a multiconductor transmission line, we can use the condition of equality of the signal amplitude at the beginning of the line $U_{\text{IN}}(t)$ and half the electromotive force of the signal source $E(t)$. Then, after simplification, we get

$$f_9 = |\max(E(t)) - 2\max(U_{\text{IN}}(t))|, \quad (25)$$

$$K_9 = \max|E(t)|.$$

This criterion must be used for $N > 2$. However, it requires the calculation of the time response at the beginning of the line, but without significant additional costs if we perform the frequency domain analysis [16].

6.5.2. Optimization of the Four-Conductor MF with Respect to Matching Criterion. A multicriteria objective function that combines one amplitude, three time, and a matching criterion (obtained for $N = 4$) as well looks (with weighing coefficients equal to 1) like

$$F = \frac{\max(U(t))}{\max(E(t))} + \frac{\tau_1 - (1/c)}{(\sqrt{\varepsilon_r \max} - 1)/c} + \frac{(\sqrt{\varepsilon_r \max}/c) - \tau_4}{(\sqrt{\varepsilon_r \max} - 1)/c} + \frac{\max(|\tau_2 - ((\tau_4 + 2\tau_1)/3)|, |\tau_3 - ((\tau_1 + 2\tau_4)/3)|)}{(\sqrt{\varepsilon_r \max} - 1)/c} + \frac{|\max(E(t)) - 2\max(U_{\text{IN}}(t))|}{\max|E(t)|}. \quad (26)$$

The MF was optimized for the following parameters: $w = 180 \mu\text{m}$, $\varepsilon_r = 5$, $l = 60 \text{ cm}$, and $R = 50 \Omega$. The value of w was unchanged, as well as the value of ε_r . The values of t , h , and s_1 , s_2 , and s_3 were optimized by the five-

TABLE 8: Optimized parameter of the MF for $N = 4$, their ranges, and values after optimization with respect to the matching criterion with equal weighing coefficients.

Parameter	Range	Value
$t, \mu\text{m}$	10–200	35
$h, \mu\text{m}$	200–2000	501
$s_1, \mu\text{m}$	1–1000	8
$s_2, \mu\text{m}$	1–1000	23
$s_3, \mu\text{m}$	1–1000	390

criterion objective function (26). The optimized parameters, their ranges, and values after optimization are given in Table 8. As a result of optimization, the amplitude was minimized to 0.0188094 V, the equalized values of $\Delta\tau_i$ were 0.53008, 0.52194, and 0.52134 ns/m, differing by 2%, and the value $(\tau_{\max} - \tau_{\min})$ was maximized to 1.57336 ns/m. In addition, the amplitude at the input MF input was 0.323928 V, which is 2.03 times less than the EMF of the source (0.657608 V), thereby ensuring the matching. The voltage waveforms at the input and output of the four-conductor MF with parameters after the GA optimization are presented in Figure 13.

Thus, in this section, we proposed and formulated a matching criterion for optimizing a multiconductor MF. Then, we obtained the five-criterion objective function for a four-conductor microstrip MF followed by the optimization of five parameters using this function. At the output of the MF, well matched to the 50Ω path, we obtained the signal with the amplitude of 0.0188094 V and close differences in the per-unit-length delays of adjacent pulses. Thus, the attenuation factor of 34.9 is possible not only for the ultrashort pulse with the duration of 65 ps considered above, but also for much longer pulses (about 100–200 ps).

Previously, we obtained a three-conductor MF for $s_1 = 10 \mu\text{m}$ and $s_2 = 115 \mu\text{m}$ with a maximum output amplitude of 0.03066 V. The optimization was carried out with respect to four criteria and four parameters in the same range. The amplitude at the MF output of this Section is 63% less.

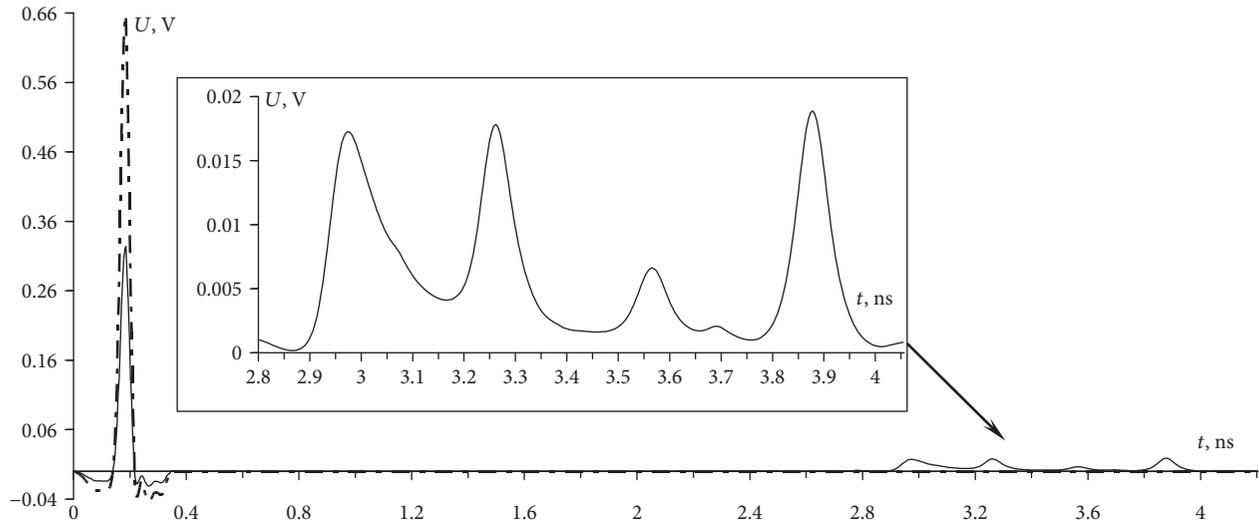


FIGURE 13: Waveforms of the EMF (---) and input (—) and output (· · ·) voltages (with enlarged fragment of the voltage waveform at the output) of a four-conductor microstrip line MF with the parameters obtained as a result of five-criterion GA optimization of five parameters.

In conclusion, we note the methodological significance of the results of this Section:

- (i) A universal five-criterion objective function is suitable for optimizing any four-conductor MF.
- (ii) The time interval criterion for a four-conductor MF is obtained in an analytical form.
- (iii) The equalization of time intervals between the decomposed pulses is approved.

These results can be successfully employed to optimize one of the new and advancing versions of a four-conductor MF called a mirror-symmetric MF [17]. In such MF, the equality of the pulse amplitudes is relatively easy to obtain, while equalizing the intervals between pulses may require optimization.

7. Conclusion

Thus, for the first time, the work presents the results of a systematic study of the possibility (both separate and simultaneous) to optimize multiconductor MFs with respect to different criteria. The formulation of the basic (electrical) optimization criteria for a MF has been performed, and analytical expressions have been proposed, which can significantly accelerate the calculation of the objective function. The amplitude and time criteria for optimizing an MF in analytical form are formulated, and a general multicriteria objective function is obtained, which allows, in the long term, to use any optimization methods and obtain higher MF characteristics. The article presents the results of optimization by heuristic search and a simple GA, and, to improve the optimization, we have formed a hybrid model consisting of heuristic search and the GA, with the formulated criteria being applicable to an MF with any N .

Note that it is useful to consider such MFs in the frequency domain and to optimize them with respect to the relevant criteria. So, for example, it is important to use an MF to protect against interference at a given frequency, as well as to provide the necessary bandwidth of a useful signal. In that case, it is possible to formulate some additional criteria that take into account the requirements in the frequency domain. In addition, it is essential to take into account the requirements to minimize mass dimensions (e.g., to protect space vehicles) as well as the cost (for large series production).

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the reviewers for valuable comments which have made it possible to considerably improve the article. This research was supported by the Ministry of Education and Science of the Russian Federation (Project 8.9562.2017/8.9).

References

- [1] Z. M. Gizatullin and R. M. Gizatullin, "Investigation of the immunity of computer equipment to the power-line electromagnetic interference," *Journal of Communications Technology and Electronics*, vol. 61, no. 5, pp. 546–550, 2016.

- [2] N. Mora, F. Vega, G. Lugrin, F. Rachidi, and M. Rubinstein, *Study and Classification of Potential IEMI Sources*, System Design and Assessment Notes, Note 41, 2014.
- [3] A. M. Zabolotsky and A. T. Gazizov, "Simulation of ultrawide band pulse propagation in asymmetrical modal filter for power network protection," *International Journal of Circuits, Systems and Signal Processing*, vol. 9, pp. 68–74, 2015.
- [4] A. T. Gazizov, A. M. Zabolotsky, and T. Rashitovich Gazizov, "UWB pulse decomposition in simple printed structures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 58, no. 4, pp. 1136–1142, 2016.
- [5] M. Reformat, E. Kuffel, D. Woodford, and W. Pedrycz, "Application of genetic algorithms for control design in power systems," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 145, no. 4, pp. 345–354, 1998.
- [6] S. K. Goudos, C. Kialalakis, and R. Mittra, "Evolutionary algorithms applied to antennas and propagation: a review of state of the art," *International Journal of Antennas and Propagation*, vol. 2016, Article ID 1010459, 12 pages, 2016.
- [7] A. O. Belousov, A. M. Zabolotsky, and T. T. Gazizov, "Experimental confirmation of the modal filtration in four- and five-conductor microstrip lines," in *2017 18th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, pp. 46–49, Erlagol, Russia, 2017.
- [8] A. O. Belousov, T. R. Gazizov, and A. M. Zabolotsky, "Maximization of duration of ultrashort pulse that is completely decomposed in multiconductor modal filters," in *2016 International Siberian Conference on Control and Communications (SIBCON)*, pp. 1–4, Moscow, Russia, 2016.
- [9] A. O. Belousov, A. M. Zabolotsky, and T. T. Gazizov, "Optimization of parameters of multiconductor modal filters for protection against ultrashort pulses," in *2016 17th International Conference of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM)*, pp. 67–70, Erlagol, Russia, 2016.
- [10] A. O. Belousov, A. M. Zabolotsky, and T. R. Gazizov, "Optimization of parameters of multiconductor modal filters for protection against ultrashort pulses," in *Materials of the all-russian scientific and technical conference «Sovremennyye problemy radioelektroniki»*, pp. 392–396, Krasnoyarsk, Russia, 2016.
- [11] A. O. Belousov, T. T. Gazizov, and T. R. Gazizov, "Multicriteria optimization of four-conductor modal filter by genetic algorithms," in *2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 445–448, Novosibirsk, Russia, 2017.
- [12] A. R. Djordjevic, R. M. Biljic, V. D. Likar-Smiljanic, and T. K. Sarkar, "Wideband frequency-domain characterization of FR-4 and time-domain causality," *IEEE Transactions on Electromagnetic Compatibility*, vol. 43, no. 4, pp. 662–667, 2001.
- [13] G. L. Matthaei and G. C. Chinn, "Approximate calculation of the high-frequency resistance matrix for multiple coupled lines," in *1992 IEEE Microwave Symposium Digest MTT-S*, pp. 1353–1354, Albuquerque, NM, USA, 1992.
- [14] S. P. Kuksenko, T. R. Gazizov, A. M. Zabolotsky et al., "New developments for improved simulation of interconnects based on method of moments," in *Proceedings of the 2015 International Conference on Modeling, Simulation and Applied Mathematics*, pp. 293–301, Phuket, Thailand, August 2015.
- [15] E. M. T. Jones, "Coupled-strip-transmission-line filters and directional couplers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 4, no. 2, pp. 75–81, 1956.
- [16] R. Achar and M. S. Nakhla, "Simulation of high-speed interconnects," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 693–728, 2001.
- [17] A. M. Zabolotsky, "Application of reflective symmetry for modal filtration improvement," *Dokladi Tomsk. gos. un-ta sist. upr. i radioelektroniki*, vol. 2, no. 36, pp. 41–44, 2015.

Research Article

An Improved MOEA/D Based on Reference Distance for Software Project Portfolio Optimization

Jing Xiao ¹, Jing-Jing Li ¹, Xi-Xi Hong¹, Min-Mei Huang ², Xiao-Min Hu ³,
Yong Tang¹ and Chang-Qin Huang ⁴

¹School of Computer Science, South China Normal University, Guangzhou, China

²School of Public Administration, South China Normal University, Guangzhou, China

³School of Computers, Guangdong University of Technology, Guangzhou, China

⁴School of Information Technology in Education, South China Normal University, Guangzhou, China

Correspondence should be addressed to Jing-Jing Li; jingjing.li1124@gmail.com

Received 29 December 2017; Revised 14 April 2018; Accepted 17 April 2018; Published 30 May 2018

Academic Editor: Jesus Medina

Copyright © 2018 Jing Xiao 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 it is becoming extremely competitive in software industry, large software companies have to select their project portfolio to gain maximum return with limited resources under many constraints. Project portfolio optimization using multiobjective evolutionary algorithms is promising because they can provide solutions on the Pareto-optimal front that are difficult to be obtained by manual approaches. In this paper, we propose an improved MOEA/D (multiobjective evolutionary algorithm based on decomposition) based on reference distance (MOEA/D_RD) to solve the software project portfolio optimization problems with optimizing 2, 3, and 4 objectives. MOEA/D_RD replaces solutions based on reference distance during evolution process. Experimental comparison and analysis are performed among MOEA/D_RD and several state-of-the-art multiobjective evolutionary algorithms, that is, MOEA/D, nondominated sorting genetic algorithm II (NSGA2), and nondominated sorting genetic algorithm III (NSGA3). The results show that MOEA/D_RD and NSGA2 can solve the software project portfolio optimization problem more effectively. For 4-objective optimization problem, MOEA/D_RD is the most efficient algorithm compared with MOEA/D, NSGA2, and NSGA3 in terms of coverage, distribution, and stability of solutions.

1. Introduction

Project portfolio management (PPM) is a management process to help project managers to analyze and acquire all information of current proposed projects. PPM helps decision makers to sort and prioritize each project according to certain criteria, such as business goals, strategic value, cost, and resource constraints. A key step of PPM is to decide which projects to invest in an optimal manner. Project portfolio optimization (PPO) is the effort to make the best decisions to select the best mix of projects from all candidate projects. Manual approaches include PPO include Q-Sort, analytic hierarchy process, and portfolio matrices [1–3]. These approaches are time-consuming and limited to the number of projects they can deal with. The project portfolio problem may be dealt as a multiobjective optimization

problem, and it is difficult to tackle [4, 5]. Software managers and researchers used branch-and-bound approach, simulated annealing and Tabu search, and so on to obtain the uniformly distributed Pareto-optimal solutions [6–8]. It is hard to find an algorithm to deal with this problem efficiently when the complexity of the problem grows exponentially with the number of projects.

Within this context, multiobjective evolutionary algorithms (MOEA) [9] which can obtain Pareto-optimal solutions are promising to solve the project portfolio optimization problem [10–12]. Pareto front-based MOEAs are superior to manual approaches in a way that they are able to create a set of efficient portfolios, for which it can be assured that there exist no solutions in the search space that promise better values in at least one of the objectives and offer at least the same values in all the other objectives [5].

MOEAs can obtain approximate optimal solutions. Furthermore, MOEAs can deal with the computational complexity with an increasing number of projects. That is why there are lots of publications devoted to solving portfolio optimization problems using MOEAs and also there are many applications of MOEAs in finance and economics areas [13].

Compared with general project portfolio optimization using MOEAs, the number of publications dedicated to the MOEAs' applications to software project portfolio problems is scarce. Kremmel et al. [5] introduced a multiobjective evolutionary approach, mPOEMS, to find the Pareto-optimal front for software project portfolio optimization problem. However, the paper only studied 2-objective optimization. In this paper, we first propose an improved MOEA/D [14] algorithm based on reference distance (MOEA/D_RD) to alleviate the inefficiency of MOEA/D's weighted sum approach. Then, we use MOEA/D_RD to solve the 2-, 3-, and 4-objective software portfolio optimization problem. Comparison and analysis experiments are conducted among MOEA/D_RD, MOEA/D, NSGA2 [15], and NSGA3 [16].

The rest of this paper is organized as follows. Section 2 discusses the related work of portfolio optimization using evolutionary algorithms. Section 3 describes the software portfolio selection model we have used. The proposed MOEA/D_RD is explained in detail in Section 4, and the empirical experiments are described and discussed in Section 5. The last section gives the conclusion and lists the future work.

2. Related Work

The first formalization of methodology for solving portfolio optimization problems was proposed by Markowitz [17] in the 1950s. Markowitz defined a portfolio as a vector of real numbers that contains the weight corresponding to each available asset and stated that the investor searches the portfolio that minimizes the risk while maximizing the return ideally. However, with the increasing number of projects and many constraints in real world, the simple assumptions in Markowitz model are infeasible and it is hard to find an exact algorithm to deal with the problem. As such, the first use of genetic algorithm (GA) for optimizing project portfolio was proposed by Arnone et al. in 1993 [18]. The authors divided the population of a GA into different subpopulations and produced different portions of the Pareto front.

An obvious advantage of MOEAs is their ability to produce, in one single run, a complete approximation of the Pareto front. MOEAs are suitable to solve the portfolio optimization problem since the aim of the problem is to provide a set of Pareto front solutions, that is, the best possible tradeoffs among the objectives, among which the managers can choose the most appropriate solution. In [19], the Markowitz model was solved with an MOEA in which the selection is carried out through a Pareto-ranking procedure. The authors used Sharpe's ratio instead of the classical density estimators such as crowding distance to break ties between solutions from the same Pareto front. Lin et al. [20] implemented integer encoding, simulated binary crossover, and parameter-based mutation within the NSGA2 to solve the investment portfolio

optimization problem with fixed transaction costs and minimum lots. Subbu et al. [21] combined a Pareto-sorting evolutionary algorithm with linear programming for investment portfolio optimization. The Pareto-sorting evolutionary algorithm is used to retain the nondominated solutions found along the search by a small population size and an archive. Branke et al. [22] combined NSGA2 with the critical line algorithm to obtain a continuous Pareto front for portfolio optimization. NSGA2 was first employed to define convex subset of the original search space, then the critical line algorithm was applied on every subset to form the complete Pareto front. Bradshaw et al. [23] employed an evolutionary algorithm similar to SPEA2 [24] to solving the portfolio optimization problem. In [25], the authors compared six MOEAs on the classical Markowitz model. The results showed that SPEA2 and NSGA2 performed more effectively among the six studied algorithms. [26] compared three MOEAs, that is, NSGA2, SPEA2, and PESA [27], to solve the Markowitz model with three objectives: return value, risk, and number of assets in the portfolio and found that SPEA2 can obtain the best performance for the test cases.

Aforementioned work is based on Markowitz mean-variance model, and there are also a few publications devoted to other portfolio optimization models using MOEAs. Khalili-Damghani et al. [28] presented a hybrid fuzzy rule-based multiobjective framework for sustainable project portfolio selection. NSGA2 was applied to obtain the non-dominated solutions. The proposed framework simultaneously considered the accuracy maximization and the complexity minimization objectives. Fernandez et al. [29] proposed a nonoutranked ant colony optimization II method for optimizing portfolio problem. The method incorporates integer linear programming to avoid clearly suboptimal regions in the search space and a priori preference system to focus the algorithmic effort on the most preferred region in the search space. Doerner et al. [4] introduced a Pareto ant colony optimization algorithm for solving the portfolio selection problem. Tofghian and Naderi [30] employed an ant colony optimization algorithm for solving the project selection and scheduling to optimize both total expected benefit and resource usage variation. Mavrotas et al. [31] studied the robustness analysis methodology for multiobjective project selection optimization.

Relatively speaking, the publication with respect to software project management using MOEAs is scarce. Rodríguez et al. [32] employed NSGA2 and a system dynamics simulation model to generate the Pareto front needed by software project managers to find the best values for initial team size and schedule estimates for a given project with the optimal cost, time, and productivity. Gueorguiev et al. [33] formulated software project planning problem as biobjective optimization. Robustness and complete time are treated as two competing objectives, and SPEA2 was employed to obtain the Pareto solutions. The most closely related to this paper is the work by Kremmel et al. [5] in which the authors used Constructive Cost Model II (COCOMO II) [34] and a multi-objective evolutionary algorithm to find the Pareto front for software project portfolio optimization. Only 2-objective Pareto front solutions were studied in Kremmel's work. In

this paper, we extend Kremmel's work and propose an improved MOEA/D algorithm called MOEA/D_RD for software project portfolio optimization. Optimization problems with 2, 3, and 4 objectives are studied using 50 projects that follow the validated COCOMO II model criteria, and the proposed approach is compared with several state-of-the-art evolutionary algorithms.

The next section presents the list of goals borrowed from Kremmel's software portfolio selection model [5]. We only use the first 4 objectives and use the synergy goal as a constraint in our framework.

3. Software Portfolio Selection Model

Generally, a multiobjective optimization problem can be presented as the following:

Find a vector $x \in \Omega$, Ω is decision (variable) space.

$$\text{maximize } \mathbf{Q}(x) = (q_1(x), q_2(x), \dots, q_n(x)), \quad \mathbf{Q} : \Omega \rightarrow R^n, \quad (1)$$

under some constraints, where Ω is the decision (variable) space, \mathbf{Q} consists of n real-valued objective functions, and R^n is called the objective space.

Suppose there are two solutions $u, v \in R^n$; u is said to dominate v if and only if $u_i \geq v_i$ for every $i \in \{1, 2, \dots, n\}$ and $u_j > v_j$ for at least one index $j \in \{1, 2, \dots, n\}$. A point $x^* \in \Omega$ is Pareto optimal to (1) if there is no point $x \in \Omega$ such that $\mathbf{Q}(x)$ dominates $\mathbf{Q}(x^*)$. $\mathbf{Q}(x^*)$ is then called a Pareto-optimal objective vector. In other words, any improvement in a Pareto-optimal point in one objective must lead to deterioration in at least one other objective. The set of all the Pareto-optimal points is called Pareto set, and the set of all the Pareto-optimal objective vectors is the Pareto front (PF).

Specifically, a solution for software project portfolio optimization is represented by a vector with the length of the maximum available projects. The task can be formalized as follows:

Find a vector $x = (x_1, x_2, \dots, x_p) \in M_1 \times \dots \times M_p$, where $M_i \subseteq M$, $M = \{0, 1, 2, \dots, T \times 12\}$, such that the objective vector $\mathbf{y} = (q_1(x), q_2(x), \dots, q_n(x))$ is maximum, where x_i is greater than 0 if project i is selected, and 0 if not; p is the number of candidate projects; M is a set of the number of months in the planning horizon; M_i is the months in which project i can start; T is the number of timeframes in the planning horizon. $q_i(x)$ is the i th optimization objective. In this work, we have considered the first 4 objectives defined in Kremmel's model, and thus the value of n is 4. These 4 objectives are defined as follows:

- (1) Potential revenue ($q_1(x)$). Software project investors invest human resources, knowledge, and money into a project, with the goal of obtaining benefits from this investment. The potential projects for the project portfolio have to be evaluated with regard to their potential financial revenue. Thus, the first objective deals with the need to maximize

potential overall portfolio return. It is calculated as the following:

$$q_1(x) = \sum_{i=1}^p r_i \cdot w_i, \quad (2)$$

where r_i is the potential revenue of project i , and w_i is 1 if $x_i > 0$ and 0 if $x_i = 0$. Obviously, the greater the overall potential revenue, the better the solution is.

- (2) Strategic alignment ($q_2(x)$). Project selection optimization has to consider the problems with little commitment from business leaders, poor alignment of projects to strategy, little coordination between projects, and conflicting project objectives. The strategic alignment on the portfolio level should be maximized. It is calculated as follows:

$$q_2(x) = \sum_{i=1}^p a_i \cdot w_i, \quad (3)$$

where a_i is the strategic alignment value of project i , and w_i is 1 if $x_i > 0$ and 0 if $x_i = 0$. The greater the overall strategic value, the better the solution is.

- (3) Resource usage distribution metric ($q_3(x)$). Resources in each timeframe are limited. This objective is to maximize the resource usage per timeframe and at the same time have the best distribution among the timeframe. Its value is between 0 and 1, where 1 means full resource consumption in each timeframe and 0 means that, at least in one timeframe, there is no resource consumed. Thus, the objective function to maximize is expressed as follows:

$$q_3(x) = \prod_{t=1}^T \left(\frac{\sum_{o=1}^l \sum_{i=1}^p r_{o,t,i} \cdot w_i}{\sum_{o=1}^l R_{o,t}} \right), \quad (4)$$

where o is the type of a resource (there are l different resource types); t is the timeframe; T is the number of timeframes in the planning horizon; $r_{o,t,i}$ is the type o resource consumption of project i in timeframe t , and $R_{o,t}$ is the type o resource limit in timeframe t . The closer the $q_3(x)$ is to one, the better the solution is.

- (4) Risk ($q_4(x)$). The risk objective is calculated as follows:

$$q_4(x) = 1 - \frac{1}{p} \sum_{i=1}^p \text{risk}_i(x) \cdot w_i, \quad (5)$$

where $\text{risk}_i(x)$ is the risk value of project i . The closer the $q_4(x)$ is to one, the better the solution is.

The constraints we have used are listed as follows:

- (a) Project starting timeframes. Most projects cannot start in an arbitrary timeframe, but very often in a few distinct timeframes. It is also possible that a

project can only start in one timeframe in order to meet a special market opportunity. A feasible solution must adhere to the constraint of project starting time.

- (b) The “must-select” restriction. Due to the legal and economic circumstances, a project may have to be included in a valid portfolio. Therefore, it should be possible to define a “must-select” restriction for portfolio optimization.
- (c) Logical relationships. There are several logical relationships between projects such as linear, dependent, and mutually exclusive relationships. Linear relationship means if a certain project is selected for a portfolio, one or more predecessor projects must be selected obligatorily. If two projects are dependent, it means that the two projects must be selected to a portfolio together. On the contrary, two projects may not be selected for the same portfolio and thus are mutually exclusive.
- (d) Synergy effects. The synergy effect constraint is one of the objectives in the original Kremmel’s model. We consider it as a constraint when we optimize the first objective, that is, the potential revenue. If two projects are selected for the same portfolio, the total revenue could be more than the sum of the two project’s revenues or less than the sum. The synergy effects are also considered in the Pareto ant colony optimization approach presented in [4].

In this paper, we consider the aforementioned four objectives and the four constraints for the software project portfolio problem. In the next section, we introduce the algorithm called MOEA/D_RD to solve the multiobjective optimization problem for software project selection.

4. An Improved MOEA/D Based on Reference Distance (MOEA/D_RD)

4.1. MOEA/D Based on Weighted Sum Approach. In this paper, we improve the weighted sum approach in MOEA/D algorithm [14] for solving the software project selection optimization problem. The approach considers a convex combination of the different objectives. Let $\lambda = (\lambda_1, \dots, \lambda_m)^T$ be a weight vector; m is the number of objectives; $f_i(x)$ is the i th objective to be optimized; and $\lambda_i \geq 0$ for all $i = 1, \dots, m$ and $\sum_{i=1}^m \lambda_i = 1$. Then the optimal solution to the following scalar optimization problem

$$\text{maximize } g^{ws}(x | \lambda) = \sum_{i=1}^m \lambda_i f_i(x) \quad \text{subject to } x \in \Omega \quad (6)$$

is a Pareto-optimal point to (1) as we can see that $q_i(x)$ corresponds to one of the objectives $f_i(x)$ in (6), where we use $g^{ws}(x | \lambda)$ to emphasize that λ is a coefficient vector in this objective function, where x is the variables to be optimized. To generate a set of different Pareto-optimal

vectors, one can use different weight vectors $\lambda^1, \lambda^2, \dots, \lambda^N$ in the above scalar optimization problem and the optimized problem is divided into N subproblems. The greater N is, the wider the search space is. However, the weighted sum-based MOEA/D has several drawbacks and we illustrate them as follows.

Given an example, as shown in Figure 1, f_1 and f_2 are two objectives; $F(x_i)$ is the objective function of solution x_i ; PF is the assumed optimal Pareto front; x_1, x_2 , and x_3 are three solutions corresponding to weight vectors λ^1, λ^2 , and λ^3 . The ideal case is that the algorithm moves x_1, x_2 , and x_3 to meet the PF. MOEA/D randomly picks up two solutions from the neighborhood of x_2 and generates a new solution using genetic operators. If the fitness value of the new solution is better than x_2 , then x_2 is replaced by the new solution. If the new solution is fallen in the overlapping area of the search spaces of neighboring solutions x_1 and x_2 , then both x_1 and x_2 are replaced by the new solution. The strategy is efficient at the earlier search stage of the algorithm, and it can make the search direction move fast to the PF. But at the late stage of the algorithm, as shown in Figure 2, there is no overlapping area among most of the search spaces of solutions. The neighboring solutions of x_2 cannot generate a new effective solution. The search process would stagnate at the late stage of the algorithm.

If the PF is a line, as shown in Figure 3, for the weight vector λ^2 , all solutions on the PF line are the same optimal solutions with the same fitness values. Among the solutions between the weight vectors λ^1 and λ^2 , the optimal solution is the intersection point of PF and f_2 . Similarly, among the solutions between the weight vectors λ^2 and λ^3 , the optimal solution is the interaction point of PF and f_1 . Assume that a solution with respect to λ^2 during iteration is x_2 , x_2 would not be replaced by x^* even if x^* is closer to λ^2 and is a better solution. It is because x_2 and x^* are equally optimal with the same fitness values on the PF line. The search process is in a standstill.

If the PF is a convex curve, as shown in Figure 4, assume that x_2 is the solution with respect to λ^2 during iteration; when the algorithm finds another solution x^* , x_2 will be replaced by x^* since the fitness value of x^* is better than x_2 . Similarly, the solutions with respect to λ^1 and λ^3 will be replaced by the solutions that are located close to the ends of PF. At the late search process of the algorithm, most of solutions are aggregated at the ends of PF and the algorithm suffers in stagnation.

From the above analysis, we can see that the traditional weighted sum approach of MOEA/D suffers poor search ability. In the next subsection, we propose an improved MOEA/D based on reference distance to enhance the search ability of the algorithm.

4.2. An Improved Algorithm MOEA/D_RD Based on Reference Distance. To alleviate the aforementioned problems of MOEA/D, we propose an improved version based on reference distance, called MOEA/D_RD. Reference distance is the distance from each solution to the weight vector,

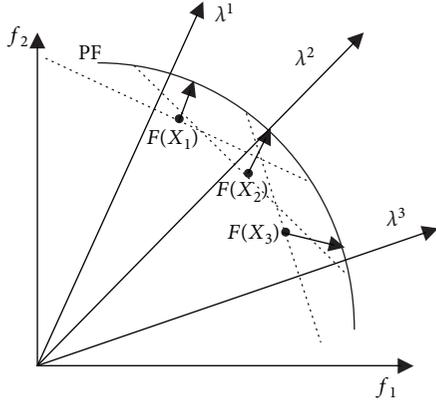


FIGURE 1: Solution generation process of MOEA/D.

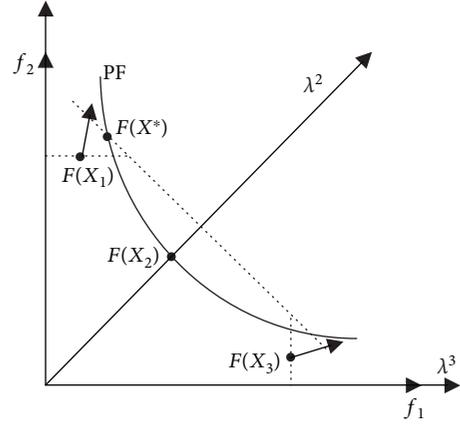


FIGURE 4: Pareto front as convex.

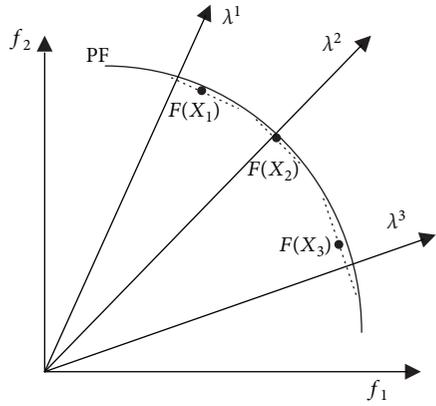


FIGURE 2: The late stage of MOEA/D.

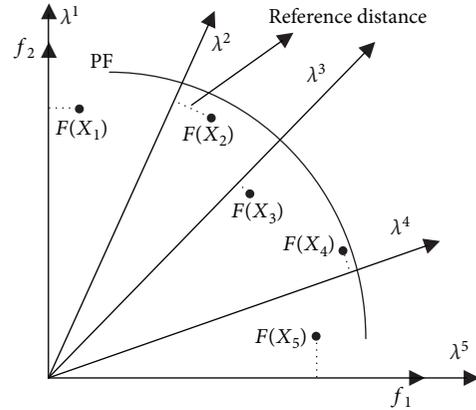


FIGURE 5: Reference distance.

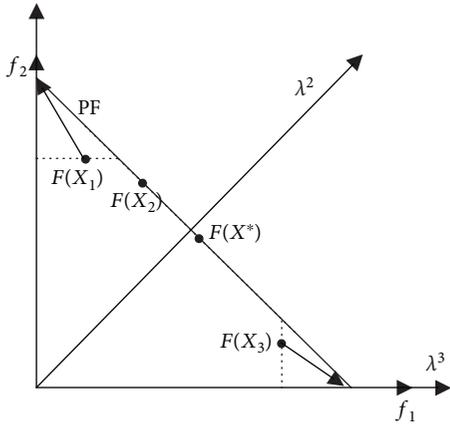


FIGURE 3: Pareto front as a line.

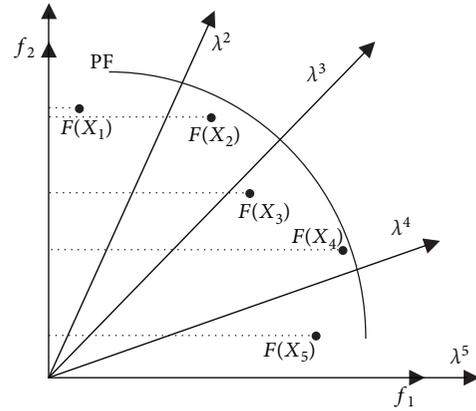


FIGURE 6: Reference distance to λ^1 .

as shown in Figure 5. For each weight vector, we can calculate the distance of all solutions to it. For example, the distance of five solutions to weight vector λ^1 is depicted in Figure 6. We can see that x_1 is the solution with the shortest distance to λ^1 among the solutions. The calculation of reference distance is described in the following.

Given: the weight vector λ , the line from original point to λ L , the solution $F(x)$, the projection point from

$F(x)$ to L is y ; then, the distance d_1 from original point to y is

$$d_1 = \frac{\|F(x)^T \lambda\|}{\|\lambda\|}, \tag{7}$$

$$\text{Reference distance } d_2 = \|F(x) - d_1 \lambda\|. \tag{8}$$

MOEA/D_RD is described as follows.

Input

(i) Multiobjective optimization fitness function, that is, (6)

(ii) R and N (iii) A uniform spread of N weight vectors: $\lambda^1, \lambda^2, \dots, \lambda^N$

Output: EP.

Step 1. Initialization:Step 1.1. Set $EP = \emptyset$, $Count = 0$;Step 1.2. Calculate the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each weight vectors. For each $i = 1, \dots, N$, set $B(i) = \{i_1, \dots, i_T\}$, where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are the T closest weight vectors to λ^i .Step 1.3. Randomly generate an initial population $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N \in \Omega$. Compute $FV^i = F(\mathbf{x}^i)$.**Step 2. Update:**For $i = 1, \dots, N$, do the following.Step 2.1. Randomly select two indexes, k and l from $B(i)$, and then generate a new solution y from \mathbf{x}^k and \mathbf{x}^l by using general genetic operators.Step 2.2. Check if y satisfies the constraints; if no, adjust y to meet the constraints and mark y to y^* .Step 2.3. Update the neighboring solutions. For each index $j \in B(i)$, if $g(y^* | \lambda) \geq g(\mathbf{x}^j | \lambda)$, then set $\mathbf{x}^j = y^*$ and $FV^j = F(y^*)$; $Count = Count + 1$.Step 2.4. Update the EP. Remove all the vectors dominated by $F(y^*)$ from EP. Add $F(y^*)$ to EP if there is no vector in EP that dominates $F(y^*)$.Step 2.5. If $Count \leq (N/R)$, go to Step 2.6; else, go to Step 3.

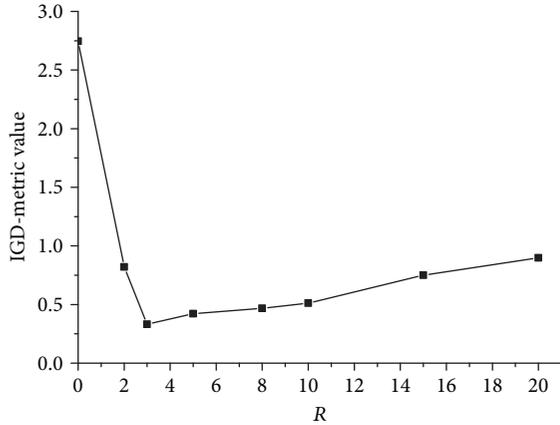
Step 2.6. Find all the subproblems where solutions are not replaced and find the corresponding weight vectors to each subproblem.

Step 2.7. Adjust the values of fitness functions for the solutions in EP and normalize them to $[0, 1]$.

Step 2.8. For each weight vector found in Step 2.6, calculate the reference distances from the solutions in EP to the vector; find the solution with the shortest distance and use it to replace the current solution with respect to the corresponding subproblem.

Step 3. Stopping Criteria: If stopping criteria is satisfied, then stop and output EP. Otherwise, go to Step 2.

ALGORITHM 1

FIGURE 7: The effect of different R values.At each generation t , MOEA/D_RD maintains

- (i) N is the number of the subproblems considered in MOEA/D_RD,
- (ii) a population of N points $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N \in \Omega$; where \mathbf{x}^i is a vector and is the current solution to the i th subproblem; \mathbf{x}^i corresponds to the weight vector λ^i ,
- (iii) FV^1, \dots, FV^N , where FV^i is the fitness value of \mathbf{x}^i , that is, $FV^i = F(\mathbf{x}^i)$ for each $i = 1, \dots, N$,
- (iv) an external population (EP), which is used to store nondominated solutions found during the search,

- (v) a variable R , $0 < R < N$; N/R stands for the replace rate; the value of R is empirically set. The variable $Count$ is used to record the number of solutions being replaced at each generation.

The algorithm works as follows.

Take Figure 5 as an example. Assume $N = 5$ and $R = 2$; if the solutions corresponding to λ^1 and λ^4 are replaced, that is, $Count = 2$, since $Count < N/R$, the corresponding solutions of $\lambda^2, \lambda^3, \lambda^5$ need to be replaced. Because x_5 is dominated by x_4 , x_5 is not in EP. x_1, x_2, x_3 , and x_4 are in EP. x_2 is used to replace the solution in terms of λ^2 since x_2 has the shortest distance to λ^2 . And x_3 is used to replace the solution in terms of λ^3 since x_3 has the shortest distance to λ^3 . Although x_5 has the shortest distance to λ^5 , but x_5 is not in EP, thus x_4 is used to replace the solution in terms of λ^5 .

From the above example and algorithm description, we can see that MOEA/D_RD has the following features:

- (1) The replacing strategy of MOEA/D_RD makes some unselected nondominate solutions in MOEA/D to generate the new population.
- (2) Although MOEA/D uses uniform weight vector, the subproblems of multiple weight vectors may fall in the same area and it may bring about the low diversity of population. MOEA/D_RD brings the idea of reference distance, and it can help the individuals that stuck in local area to search more widely.

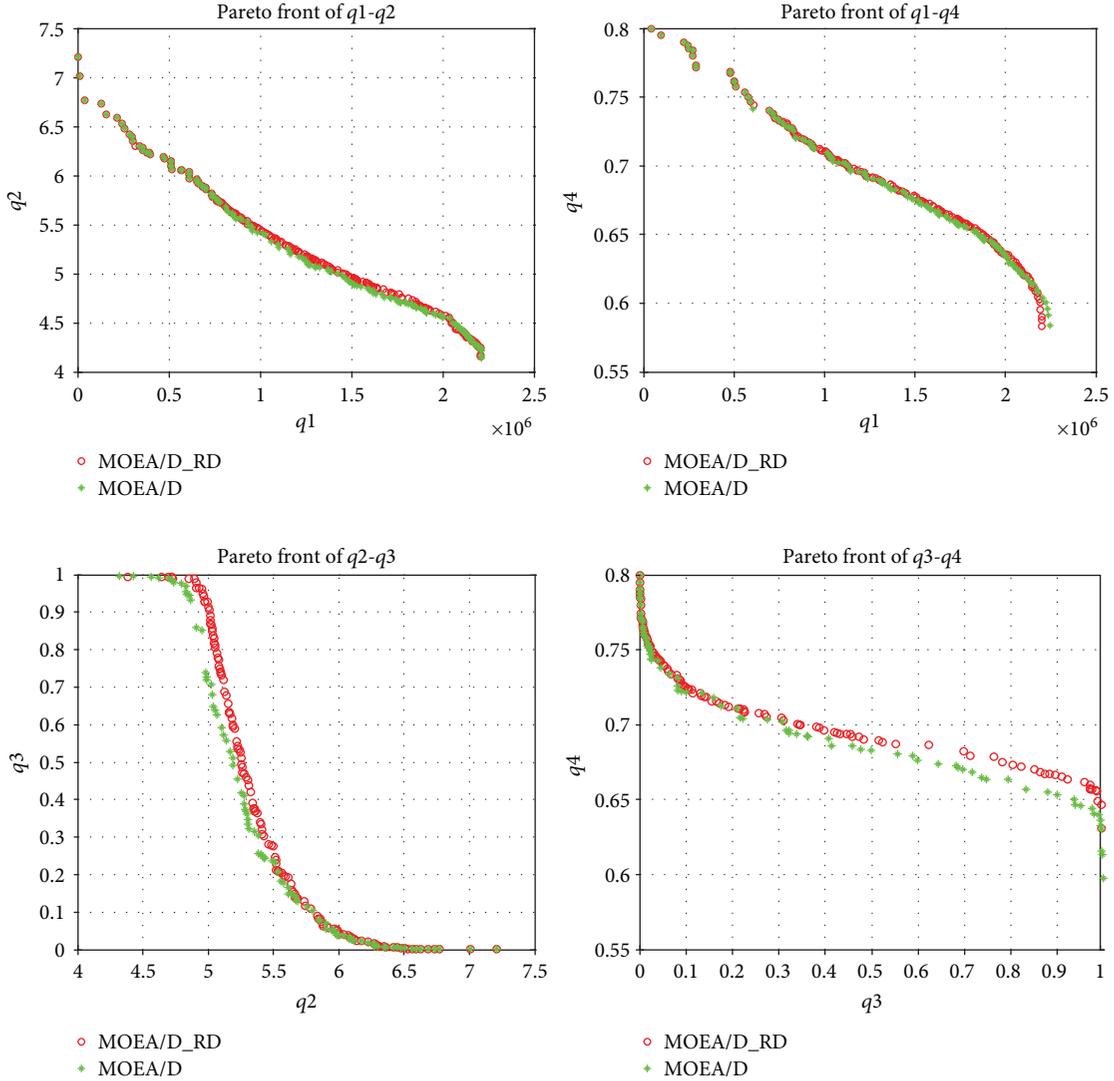


FIGURE 8: Comparison of PF between MOEA/D_RD (red circle) and MOEA/D (green *) for 2-objective optimization.

- (3) MOEA/D_RD can bring new individuals when the algorithm is in stagnation; at the same time, the reference distance can guarantee that the new individuals are generated from the parents in neighborhood.

In brief, compared to the original MOEA/D, the replacing strategy based on reference distance in MOEA/D_RD increases the diversity of population and can obtain well-distributed solutions. The improved algorithm performs well in high-dimensional multiobjective optimization.

5. Experimental Evaluation

This section presents the experiments carried out to evaluate the performance of the proposed approach. First, the test data set based on the Constructive Cost Model (COCOMO II) is described. Three evaluating metrics are then introduced. Lastly, we compare MOEA/D_RD with MOEA/D, NSGA2, and NSGA3. All experiments were run on an Intel

Core i5-2450M CPU@2.50 GHz, 4GB memory PC with Win7 64-bit operating system.

5.1. COCOMO II Test Set. COCOMO II is a model to estimate the cost, effort, and schedule when planning a new software development activity. The test set is based on this model and consists of 50 software projects [5]. The number of lines of source code of these projects is between 1000 and 37000. The maximum duration of a project is 18 months, and the planning horizon is set to 3 years. The planning horizon is divided into 3 timeframes, one year (12 months) per timeframe. There are 1500 person-months in total for the planning horizon and 500 person-months per timeframe. Each project has an assigned risk value between 0.2 and 0.8. Potential revenue is set to the maximum of 150% and to the minimum of 85% of the initial costs. The total strategic alignment value is calculated by a weight sum of each strategy's alignment value which is set randomly. A maximum

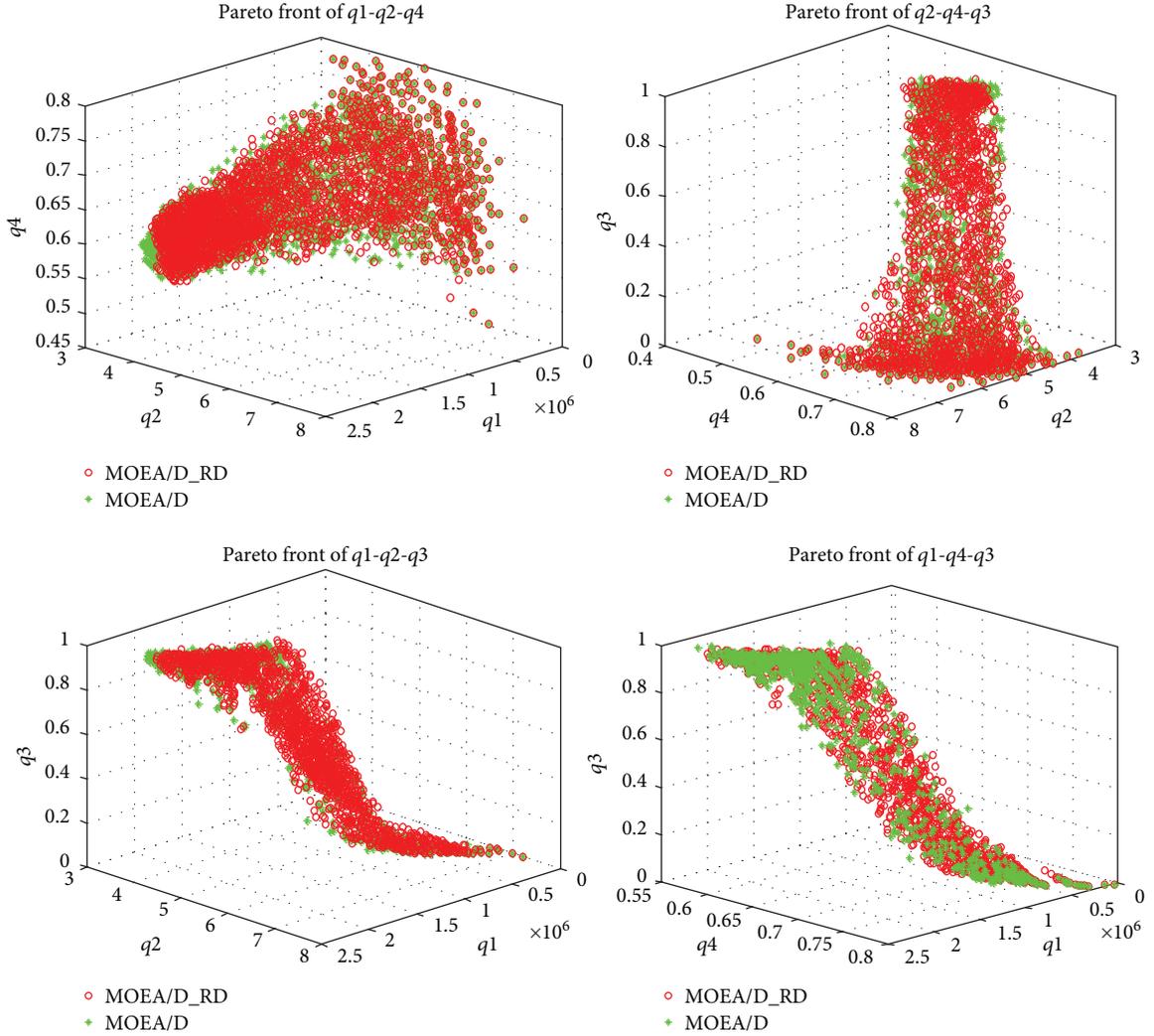


FIGURE 9: Comparison of PF between MOEA/D_RD (red circle) and MOEA/D (green *) for 3-objective optimization.

number of 30% of all projects are selected to have synergy effects with exactly one project where there is 15% of the positive synergy and 15% of the negative synergy. A number of 10% of all projects are selected randomly to be mandatory, and 4 projects are manually selected to be mutually exclusive.

5.2. Evaluation Metrics. In order to verify the proposed algorithm and compare to other state-of-the-art algorithms, we use three performance indexes as the following:

- (i) Set coverage (C-metric) [14]: Let A and B be the two approximations to the PF of a multiobjective optimization problem, $C(A, B)$ is defined as the percentage of the solutions in B that are dominated by at least one solution in A , that is,

$$C(A, B) = \frac{|\{u \in B \mid \exists v \in A : v \text{ dominates } u\}|}{|B|}. \quad (9)$$

$C(A, B) = 1$ means that all solutions in B are dominated by some solutions in A , while $C(A, B) = 0$ implies that no solution in B is dominated by a solution in A .

- (ii) IGD-metric [7]: Let A be a set of nondominated solutions obtained by the algorithm. Let P^* be the true PF. Since we do not know the actual PF for the software portfolio optimization problem in this paper, we use the optimal solutions obtained by all the compared algorithms as the approximation of P^* . The average distance from P^* to A is defined as

$$\text{IGD}(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}, \quad (10)$$

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A . If $|P^*|$ is large enough to represent the PF very well, $\text{IGD}(A, P^*)$ could

TABLE 1: Comparison of C-metric of MOEA/D_RD and MOEA/D.

	C-metric (MOEA/D_RD, MOEA/D)	C-metric (MOEA/D, MOEA/D_RD)
q_1 - q_2	0.559	0.071
q_1 - q_4	0.527	0.0628
q_2 - q_3	0.768	0.021
q_3 - q_4	0.75	0.018
q_1 - q_2 - q_3	0.825	0.005
q_1 - q_2 - q_4	0.265	0.321
q_1 - q_4 - q_3	0.81	0.01
q_2 - q_4 - q_3	0.632	0.063
q_1 - q_2 - q_3 - q_4	0.43	0.176

TABLE 2: Comparison of IGD-metric of MOEA/D_RD and MOEA/D.

	IGD-metric (MOEA/D)	IGD-metric (MOEA/D_RD)
q_1 - q_2	3669.74	313.09
q_1 - q_4	4273.46	818.957
q_2 - q_3	0.037	0.001
q_3 - q_4	0.007	0.001
q_1 - q_2 - q_3	1701.49	46.68
q_1 - q_2 - q_4	212.82	544.87
q_1 - q_4 - q_3	1732.64	31.49
q_2 - q_4 - q_3	0.021	0.002
q_1 - q_2 - q_3 - q_4	100.86	57.68

TABLE 3: Comparison of GD-metric of MOEA/D_RD and MOEA/D.

	GD-metric (MOEA/D)		GD-metric (MOEA/D_RD)	
	Mean	Deviation	Mean	Deviation
q_1 - q_2	4361.43	5017.01	2751.47	3548.01
q_1 - q_4	4364.78	5496.39	2435.11	3272.23
q_2 - q_3	0.062	0.063	0.043	0.058
q_3 - q_4	0.013	0.011	0.011	0.011
q_1 - q_2 - q_3	1741.47	2734.43	662.37	1422.7
q_1 - q_2 - q_4	338.23	503.61	310.92	459.31
q_1 - q_4 - q_3	1591.24	3665.13	559.19	1084.4
q_2 - q_4 - q_3	0.022	0.02	0.018	0.016
q_1 - q_2 - q_3 - q_4	116.19	215.604	142.091	213.107

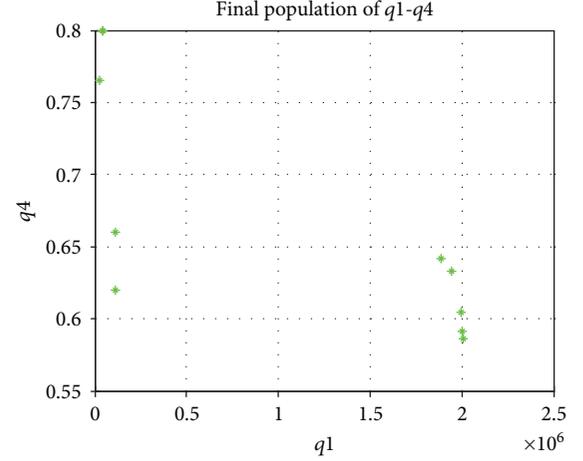


FIGURE 10: The final population of MOEA/D.

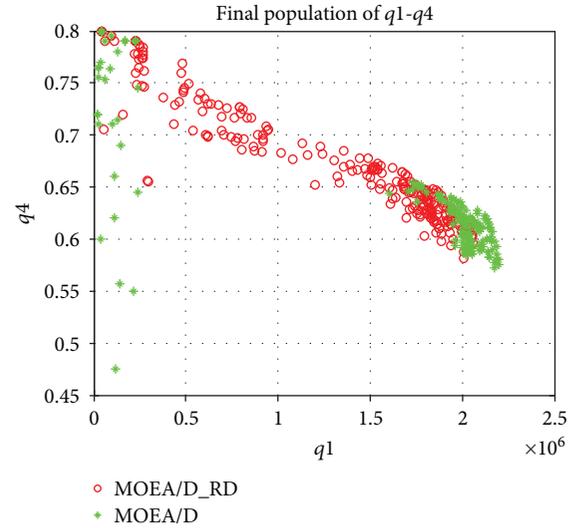


FIGURE 11: The final population of MOEA/D_RD and MOEA/D.

measure both the diversity and convergence of A in a sense. To have a low value of $IGD(A, P^*)$, set A must be very close to the PF.

(iii) GD-metric [7]:

$$GD(A, P^*) = \frac{\sum_{v \in A} d(v, P^*)}{|A|}, \quad (11)$$

where P^* and A have the same definitions as in IGD-metric. $d(v, P^*)$ is the minimum Euclidean distance between v and the points in P^* . The smaller value GD-metric is, the more stable the algorithm.

To decide the value of R in MOEA/D_RD suitably, we first analyze the impact of R to the performance. Figure 7 shows the different IGD-metric values with different R values in the objectives of q_2 and q_3 optimization task. For every R value, the algorithm is run 20 times and the reference PF consists of all the nondominated solutions of the 20 runs.

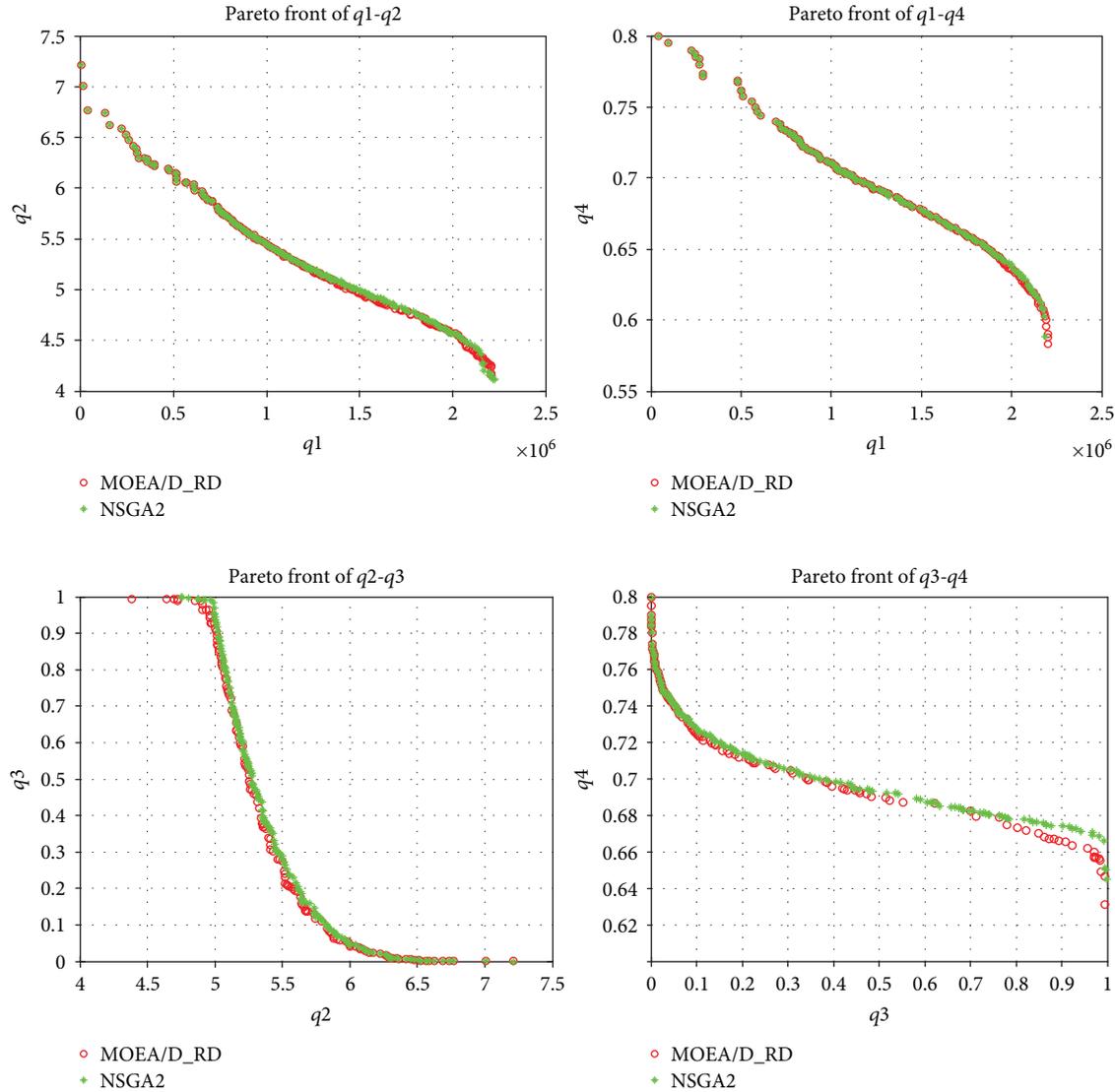


FIGURE 12: Comparison of PF between MOEA/D_RD (red circle) and NSGA2 (green *) for 2-objective optimization.

Calculate the IGD-metric for every single R value. When R equals to 0, it means reference distance is not used in the algorithm. We can see that the IGD-metric is the best when R is 3. The algorithm performs similarly when R is between 3 and 8. Considering that the convergence is slow if R is too small and the stagnation in search process is serious if R is too large, we set R to 5 in the following experiments.

5.3. Comparison between MOEA/D_RD and MOEA/D. As for the 4 objectives we mentioned in Section 3, q_1 and q_3 are positively correlated; that is, high revenue can be expected only when resources are effectively used throughout the whole planning horizon and vice versa. For q_2 and q_4 , usually the project with either the lowest risk or the highest strategic alignment value is selected to the portfolio. Thus, q_1 - q_3 and q_2 - q_4 are not studied in our 2-objective optimization experiments. The experiments are conducted on 2-objective optimization problems: q_1 - q_2 , q_1 - q_4 , q_2 - q_3 , and q_3 - q_4 ; 3-

objective optimization problems: q_1 - q_2 - q_4 , q_1 - q_2 - q_3 , q_1 - q_3 - q_4 , and q_2 - q_3 - q_4 ; and 4-objective optimization problem: q_1 - q_2 - q_3 - q_4 . There are 150 weight vectors in 2-objective optimization experiments, 351 weight vectors in 3-objective optimization experiments, and 455 weight vectors in 4-objective optimization experiment. The number of neighborhood is 10. The mutation rate is 0.01. The number of generation is 500 for 2-objective optimization and 1000 for 3- and 4-objective optimization problems. We run 20 independent runs with each of the compared algorithms where each run produced a set of nondominated solutions. The final population of nondominated solutions is plotted in Figures 8 and 9. We can see that MOEA/D-RD obtains more nondominated solutions.

Tables 1–3 give the comparisons between MOEA/D-RD and MOEA/D in terms of C-metric, IGD-metric, and GD-metric. The better performance is marked in bold. From Tables 1 and 2, we can see that MOEA/D outperforms

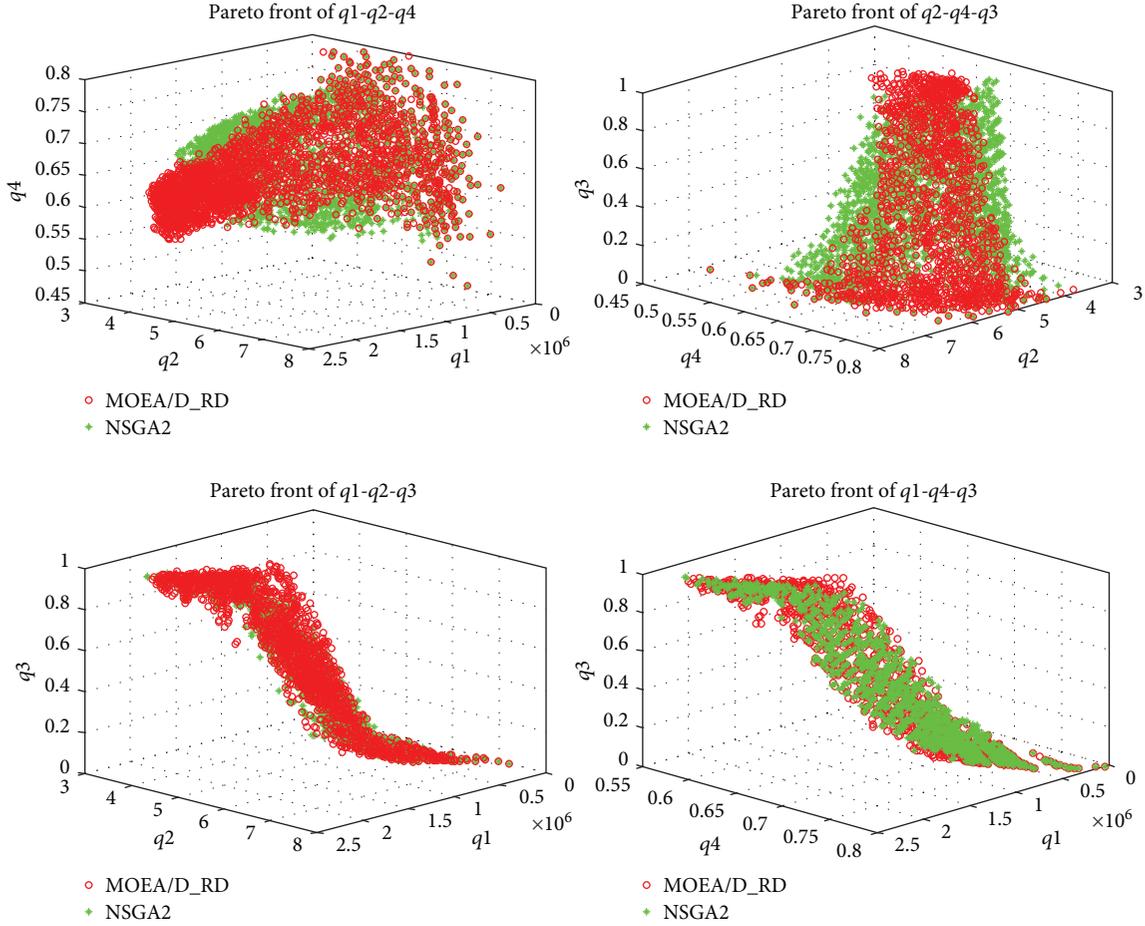


FIGURE 13: Comparison of PF between MOEA/D_RD (red circle) and NSGA2 (green *) for 3-objective optimization.

TABLE 4: Comparison of C-metric of MOEA/D_RD and NSGA2.

	C-metric (MOEA/D_RD, NSGA2)	C-metric (NSGA2, MOEA/D_RD)
q_1 - q_2	0.048	0.44
q_1 - q_4	0.036	0.188
q_2 - q_3	0	0.78
q_3 - q_4	0	0.702
q_1 - q_2 - q_3	0.027	0.531
q_1 - q_2 - q_4	0.183	0.15
q_1 - q_4 - q_3	0.024	0.385
q_2 - q_4 - q_3	0.052	0.524
q_1 - q_2 - q_3 - q_4	0.24	0.14

TABLE 5: Comparison of IGD-metric of MOEA/D_RD and NSGA2.

	IGD-metric (NSGA2)	IGD-metric (MOEA/D_RD)
q_1 - q_2	313.793	2024.71
q_1 - q_4	427.488	619.194
q_2 - q_3	0	0.0127
q_3 - q_4	0	0.006
q_1 - q_2 - q_3	189.272	324.263
q_1 - q_2 - q_4	40949.7	207.512
q_1 - q_4 - q_3	138.696	326.496
q_2 - q_4 - q_3	0.008	0.026
q_1 - q_2 - q_3 - q_4	6633.18	67.3109

MOEA/D-RD in only one item of q_1 - q_2 - q_4 . MOEA/D_RD performs better in the other 8 optimization problems with smaller C-metric and IGD-metric.

Figures 10 and 11 illustrate the improvement of the diversity of population and the distribution uniformity of solutions in MOEA/D_RD. Figure 10 presents the final population in one random run for q_1 - q_4 problem. The number of

population is 150, and there are only 9 different solutions at the last generation. We can see that the solutions in the neighborhood are almost the same and there is no new solution generated through genetic operators. The algorithm suffers in stagnation. Figure 11 gives the final population after 20 runs for q_1 - q_4 problem. We can see that the nondominated solutions obtained by MOEA/D_RD are distributed more uniformly than the solutions by MOEA/D.

TABLE 6: Comparison of GD-metric of MOEA/D_RD and NSGA2.

	GD-metric (NSGA2)		GD-metric (MOEA/D_RD)	
	Mean	Deviation	Mean	Deviation
q_1 - q_2	957.144	2152.5	2751.47	3548.01
q_1 - q_4	1223.3	2475.81	2435.11	3272.23
q_2 - q_3	0.003	0.011	0.043	0.058
q_3 - q_4	0.002	0.003	0.011	0.011
q_1 - q_2 - q_3	591.726	2347.86	662.374	1422.7
q_1 - q_2 - q_4	228.112	437.182	310.917	459.31
q_1 - q_4 - q_3	459.115	1074.91	559.196	1084.4
q_2 - q_4 - q_3	0.01	0.011	0.018	0.016
q_1 - q_2 - q_3 - q_4	184.506	324.817	142.091	213.107

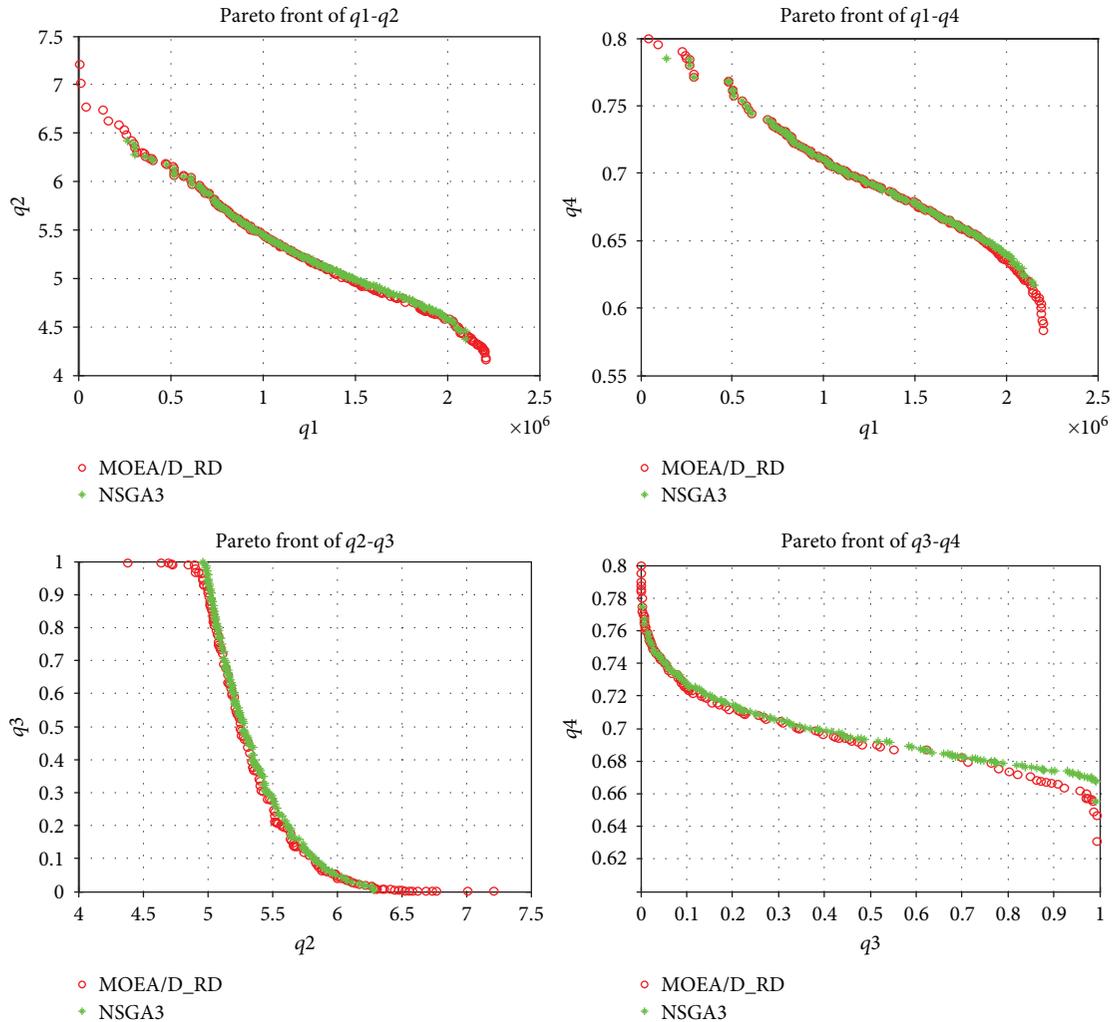


FIGURE 14: Comparison of PF between MOEA/D_RD (red circle) and NSGA3 (green *) for 2-objective optimization.

5.4. Comparison between MOEA/D_RD and NSGA2. NSGA2 has performed effectively in various optimization problems since it is invented. We also employ NSGA2 to solve our software project portfolio optimization model. The number of

population of NSGA2 is 160, 360, and 500, respectively, for 2-objective, 3-objective, and 4-objective optimization problems. The mutation rate is 0.01. The number of generation is 500 for 2-objective optimization problem and 1000 for

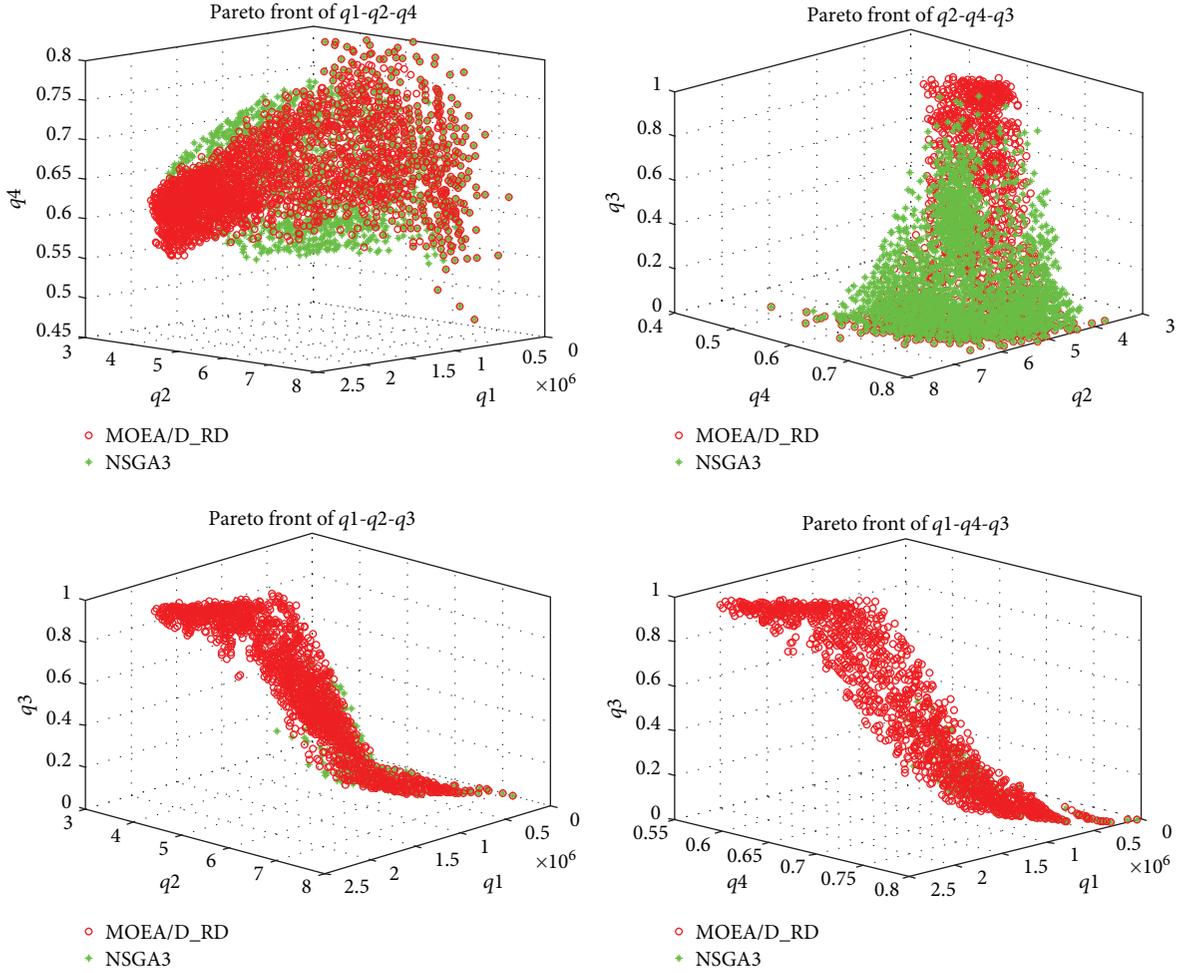


FIGURE 15: Comparison of PF between MOEA/D_RD (red circle) and NSGA3 (green *) for 3-objective optimization.

3- and 4-objective optimization problems. The final population of nondominated solutions is plotted in Figures 12 and 13 after 20 independent runs. We can see that NSGA2 performs better and obtains more nondominated solutions in 2- and 3-objective optimization problems. Tables 4–6 give the comparisons between MOEA/D-RD and NSGA2 in terms of C -metric, IGD-metric, and GD-metric. The better performance is marked in bold. MOEA/D_RD outperforms NSGA2 for 4-objective optimization problem.

5.5. Comparison between MOEA/D_RD and NSGA3. NSGA3 is the newest version of NSGA2 designed for many objective optimization problems. We use NSGA3 to solve the software project portfolio optimization problem especially the 3- and 4-objective optimization problems. The number of population for NSGA3 is set to 152, division parameter ρ 150, reference points 151 for 2-objective optimization experiments. The number of population is set to 352, division parameter ρ 25, reference points 351 for 3-objective optimization experiments. The number of population is 456, division parameter ρ 12, reference points 455 for 4-objective optimization experiments. The mutation rate is 0.01. The number of generation is 500 for biobjective optimization and 1000 for 3- and 4-

objective optimization problems. The final population of nondominated solutions is plotted in Figures 14 and 15 after 20 independent runs. For 2-objective optimization problem, MOEA/D_RD can obtain more nondominated solutions and the distribution of solutions is more uniformly than NSGA3. For 3-objective optimization problem, the solutions obtained by NSGA3 are distributed in smaller area than MOEA/D_RD.

Tables 7–9 give the comparisons between MOEA/D_RD and NSGA3 in terms of C -metric, IGD-metric, and GD-metric. The better performance is marked in bold. From Tables 7 and 8, we can see that NSGA3 outperforms MOEA/D_RD in terms of C -metric while MOEA/D_RD outperforms NSGA3 in terms of IGD-metric. It means that NSGA3 can get better PF, but the distribution of solutions is worse than MOEA/D_RD. Table 9 indicates that NSGA3 can have better mean values but worse deviation values than MOEA/D_RD and we can have the same conclusion from Tables 7 and 8.

5.6. More Experiments. We also conducted some experiments to compare the four studied algorithms. The results using the PF obtained by the four algorithms and the correspondingly

TABLE 7: Comparison of C-metric of MOEA/D_RD and NSGA3.

	C-metric (MOEA/D_RD, NSGA3)	C-metric (NSGA3, MOEA/D_RD)
q_1 - q_2	0.02	0.422
q_1 - q_4	0.005	0.285
q_2 - q_3	0.024	0.773
q_3 - q_4	0.007	0.667
q_1 - q_2 - q_3	0.144	0.065
q_1 - q_2 - q_4	0.194	0.142
q_1 - q_4 - q_3	0.156	0.055
q_2 - q_4 - q_3	0.009	0.498
q_1 - q_2 - q_3 - q_4	0.083	0.107

TABLE 8: Comparison of IGD-metric of MOEA/D_RD and NSGA3.

	IGD-metric (NSGA3)	IGD-metric (MOEA/D_RD)
q_1 - q_2	8664.81	2028.22
q_1 - q_4	2556.45	1077.42
q_2 - q_3	0.027	0.013
q_3 - q_4	0.001	0.006
q_1 - q_2 - q_3	156706	43.468
q_1 - q_2 - q_4	4337.33	179.436
q_1 - q_4 - q_3	203970	54.698
q_2 - q_4 - q_3	0.013	0.019
q_1 - q_2 - q_3 - q_4	125345	81.746

TABLE 9: Comparison of GD-metric of MOEA/D_RD and NSGA3.

	GD-metric (NSGA3)		GD-metric (MOEA/D_RD)	
	Mean	Deviation	Mean	Deviation
q_1 - q_2	517.813	2271.78	2751.47	3548.01
q_1 - q_4	746.94	1970.08	2435.11	3272.23
q_2 - q_3	0.002	0.014	0.043	0.058
q_3 - q_4	0.001	0.002	0.011	0.011
q_1 - q_2 - q_3	258.735	1916.26	662.374	1422.7
q_1 - q_2 - q_4	249.137	594.39	310.917	459.31
q_1 - q_4 - q_3	463.686	4324.19	559.196	1084.4
q_2 - q_4 - q_3	0.006	0.016	0.018	0.016
q_1 - q_2 - q_3 - q_4	67.996	393.873	142.091	213.107

computed IGD-metric are shown in Table 10. NSGA2 performs the best in 6 2- and 3-objective optimization problems with the smallest IGD-metric values. But for the 4-objective optimization problem, MOEA/D_RD outperforms the other algorithms.

TABLE 10: IGD-metric of the compared four algorithms.

	MOEA/D_RD	MOEA/D	NSGA2	NSGA3
q_1 - q_2	2327.95	4616.15	844.444	9693.75
q_1 - q_4	1771.3	4314.13	1924.44	3258.74
q_2 - q_3	0.013	0.056	0	0.027
q_3 - q_4	0.007	0.011	0.001	0.001
q_1 - q_2 - q_3	334.079	1908.11	222.284	120820
q_1 - q_2 - q_4	615.151	341.423	39321.5	5671.34
q_1 - q_4 - q_3	344	1893.54	138.744	148423
q_2 - q_4 - q_3	0.026	0.04	0.012	0.019
q_1 - q_2 - q_3 - q_4	136.089	146.714	5616.01	113234

TABLE 11: Average running time of the four compared algorithms.

	MOEA/D_RD	MOEA/D	NSGA2	NSGA3
q_1 - q_2	1.7	1.56	19.26	52.74
q_1 - q_4	1.75	1.68	23.05	50.14
q_2 - q_3	0.97	1.11	18.21	49.05
q_3 - q_4	1.54	2.11	21.68	50.21
q_1 - q_2 - q_3	14.62	14.75	102.95	516.95
q_1 - q_2 - q_4	10.54	10.45	47.16	445.15
q_1 - q_4 - q_3	19.61	19.01	136.11	493.5
q_2 - q_4 - q_3	10.94	10.76	80.32	460.74
q_1 - q_2 - q_3 - q_4	34.28	33.25	106.57	934.84

Table 11 presents the average running time of the four algorithms in 20 runs for the 9 optimization problems. We can see that MOEA/D consumes the least time and MOEA/D_RD costs the similar time compared with MOEA/D. NSGA2 needs several or more than ten times than MOEA/D_RD, and NSGA3 consumes the most time. To sum up, MOEA/D needs the least time but it is easy to suffer in stagnation and cannot obtain good nondominated solutions. NSGA3 is not likely suitable to solve the software project portfolio optimization problem. It needs the most time, and the distribution of solutions is not good. NSGA2 performs well in 2- and 3-objective optimization problems, but it needs much more running time than MOEA/D_RD. Generally speaking, MOEA/D_RD has the excellent overall performance. It can get the uniformly distributed nondominated solutions and performs the best for the 4-objective optimization problem. Although MOEA/D_RD is a little worse than NSGA2 in 2- and 3-objective optimization problems, but it consumes much less running time. In conclusion, we can say that MOEA/D_RD is an effective approach to the software project portfolio optimization problem.

5.7. *Conclusion and Future Work.* Based on Kremmel's model, a compact model with 4-objective optimization model for software project portfolio problem is proposed in this paper. The model is adaptive for software companies

who can revise the objectives and constraints according to their own requirements. To solve the proposed model, an improved MOEA/D algorithm called MOEA/D_RD based on reference distance is proposed accordingly. The algorithm uses reference distance to select some solutions to generate new solutions. Compared with MOEA/D, NSGA2, and NSGA3, MOEA/D_RD performs well in terms of the quality of solutions and running time, especially for 4-objective optimization problem. Future work could cover several topics. A great number of projects for the test set could be conducted. It also would be important to test the approach on a test set with real-world data which may include some incomplete and uncertain data.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (NSFC) Project nos. 61202296, 71102146, 61272067, 61370229, and 61370178, the National High-Technology Research and Development Program (“863” program) of China under Grant no. 2013AA01A212, the National Science Foundation of Guangdong Province Project no. S2012030006242, the Frontier and Key Technology Innovation Special Funds in Guangdong Province Project no. 2015B010109003, and the National Key Technology R&D Program of China Project no. 2013BAH72B01.

References

- [1] S. Altuntas and T. Dereli, “A novel approach based on DEMATEL method and patent citation analysis for prioritizing a portfolio of investment projects,” *Expert Systems with Applications*, vol. 42, no. 3, pp. 1003–1012, 2015.
- [2] N. P. Archer and F. Ghasemzadeh, “An integrated framework for project portfolio selection,” *International Journal of Project Management*, vol. 17, no. 4, pp. 207–216, 1999.
- [3] J. Liu, X. Jin, T. Wang, and Y. Yuan, “Robust multi-period portfolio model based on prospect theory and ALMV-PSO algorithm,” *Expert Systems with Applications*, vol. 42, no. 20, pp. 7252–7262, 2015.
- [4] K. Doerner, W. J. Gutjahr, R. F. Hartl, C. Strauss, and C. Stummer, “Pareto ant colony optimization: a metaheuristic approach to multiobjective portfolio selection,” *Annals of Operations Research*, vol. 131, no. 1–4, pp. 79–99, 2004.
- [5] T. Kremmel, J. Kubalik, and S. Biffel, “Software project portfolio optimization with advanced multiobjective evolutionary algorithms,” *Applied Soft Computing*, vol. 11, no. 1, pp. 1416–1426, 2011.
- [6] C. H. Papadimitriou and M. Yannakakis, “On the approximability of trade-offs and optimal access of Web sources,” in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pp. 86–92, Redondo Beach, CA, USA, 2000, IEEE.
- [7] P. M. Pardalos, I. Steponavičė, and A. Žilinskas, “Pareto set approximation by the method of adjustable weights and successive lexicographic goal programming,” *Optimization Letters*, vol. 6, no. 4, pp. 665–678, 2012.
- [8] I. Radziukynienė and A. Žilinskas, “Evolutionary methods for multi-objective portfolio optimization,” in *Proceedings of the World Congress on Engineering 2008*, vol. 2, pp. 1155–1159, London, UK, 2008.
- [9] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang, “Multiobjective evolutionary algorithms: a survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [10] H.-G. Beyer, S. Finck, and T. Breuer, “Evolution on trees: on the design of an evolution strategy for scenario-based multi-period portfolio optimization under transaction costs,” *Swarm and Evolutionary Computation*, vol. 17, pp. 74–87, 2014.
- [11] C. Lin and M. Gen, “Multi-criteria human resource allocation for solving multistage combinatorial optimization problems using multiobjective hybrid genetic algorithm,” *Expert Systems with Applications*, vol. 34, no. 4, pp. 2480–2490, 2008.
- [12] M. Rabbani, M. Aramoon Bajestani, and G. Baharian Khoshkhou, “A multi-objective particle swarm optimization for project selection problem,” *Expert Systems with Applications*, vol. 37, no. 1, pp. 315–321, 2010.
- [13] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, “A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.
- [14] Q. Zhang and H. Li, “MOEA/D: a multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [16] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [17] H. Markowitz, “Portfolio selection,” *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [18] S. Arnone, A. Loraschi, and A. G. B. Tettamanzi, “A genetic approach to portfolio selection,” *Neural Network World*, vol. 3, no. 6, pp. 597–604, 1993.
- [19] S. C. Chiam, K. C. Tan, and A. Al Mamum, “Evolutionary multi-objective portfolio optimization in practical context,” *International Journal of Automation and Computing*, vol. 5, no. 1, pp. 67–80, 2008.
- [20] D. Lin, S. Wang, and H. Yan, “A multiobjective genetic algorithm for portfolio selection problem,” in *Proceedings of ICOTA 2001*, pp. 567–574, Hong Kong, 2001.
- [21] R. Subbu, P. Bonissone, N. Eklund, K. Bollapragada, and S. Chalermkraivuth, “Multiobjective financial portfolio design: a hybrid evolutionary approach,” in *2005 IEEE Congress on Evolutionary Computation*, pp. 1722–1729, Edinburgh, UK, 2005, IEEE.
- [22] J. Branke, B. Scheckenbach, M. Stein, K. Deb, and H. Schmeck, “Portfolio optimization with an envelope-based multi-objective evolutionary algorithm,” *European Journal of Operational Research*, vol. 199, no. 3, pp. 684–693, 2009.

- [23] N. Bradshaw, C. Walshaw, C. Ierotheou, and A. Parrot, "A multi-objective evolutionary algorithm for portfolio optimization," in *Proceedings of the Symposium Evolutionary Systems*, pp. 27–32, Edinburgh, UK, 2009.
- [24] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pp. 95–100, Athens, Greece, 2002.
- [25] I. Radziukyniene and A. Zilinskas, "Approximation of Pareto set in multi objective portfolio optimization," in *Advances in Electrical Engineering and Computational Science*, vol. 39 of Lecture Notes in Electrical Engineering, pp. 551–562, Springer, Dordrecht, Netherlands, 2009.
- [26] K. P. Anagnostopoulos and G. Mamanis, "A portfolio optimization model with three objectives and discrete variables," *Computers & Operations Research*, vol. 37, no. 7, pp. 1285–1297, 2010.
- [27] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," in *International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pp. 839–848, Paris, France, 2000.
- [28] K. Khalili-Damghani, S. Sadi-Nezhad, F. H. Lotfi, and M. Tavana, "A hybrid fuzzy rule-based multi-criteria framework for sustainable project portfolio selection," *Information Sciences*, vol. 220, pp. 442–462, 2013.
- [29] E. Fernandez, C. Gomez, G. Rivera, and L. Cruz-Reyes, "Hybrid metaheuristic approach for handling many objectives and decisions on partial support in project portfolio optimisation," *Information Sciences*, vol. 315, pp. 102–122, 2015.
- [30] A. A. Tofghian and B. Naderi, "Modeling and solving the project selection and scheduling," *Computers & Industrial Engineering*, vol. 83, pp. 30–38, 2015.
- [31] G. Mavrotas, J. R. Figueira, and E. Siskos, "Robustness analysis methodology for multi-objective combinatorial optimization problems and application to project selection," *Omega*, vol. 52, pp. 142–155, 2015.
- [32] D. Rodríguez, M. Ruiz, J. C. Riquelme, and R. Harrison, "Multiobjective simulation optimisation in software project management," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation - GECCO '11*, pp. 1883–1890, Dublin, Ireland, 2011.
- [33] S. Gueorguiev, M. Harman, and G. Antoniol, "Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation - GECCO '09*, pp. 1673–1680, Montreal, QC, Canada, 2009.
- [34] B. Boehm, *Software Cost Estimation with COCOMO II*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000.

Research Article

Improved Method for Predicting the Performance of the Physical Links in Telecommunications Access Networks

Ferenc Lilik ¹, Szilvia Nagy ², and László T. Kóczy ^{2,3}

¹Department of Telecommunications, Széchenyi István University, Egyetem tér 1, Győr 9026, Hungary

²Széchenyi István University, Győr, Hungary

³Budapest University of Technology and Economics, Budapest, Hungary

Correspondence should be addressed to Ferenc Lilik; lilikf@sze.hu

Received 29 December 2017; Accepted 22 March 2018; Published 16 May 2018

Academic Editor: Michele Scarpiniti

Copyright © 2018 Ferenc Lilik 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 novel approach is presented which is able to predict the available maximal data transfer rate of SHDSL connections from measured frequency dependent electrical parameters of wire pairs. Predictions are made by a fuzzy inference system. The basis of the operable and tested method will be introduced, then an improved version is shown, in which the problems derived from sampling of continuous functions of electrical parameters are eliminated by wavelet transformation. Also possibilities for simplification of the problem and a way of reducing the dimensions of the applied rule bases are presented. As the set of the measured data leads to sparse rule bases, handling of sparseness is unavoidable. Two different ways—fuzzy interpolation and various membership functions—will be introduced. The presented methods were tested by measurements in real telecommunications access networks.

1. Introduction

In telecommunications the emphasis moved from voice connections to data communications in the past decades. Despite the modern and efficient types of networks for data transmission, for example, fiber optics, huge amount of copper wire pair based networks developed for voice communications is still in use. These copper networks, especially in rural or less densely populated areas, will be in use at least in the next couple of decades.

Further use of copper networks is promoted by the development of such new methods, which aim to serve modern telecommunications services of acceptable quality on these networks. The various xDSL technologies were and are being developed to establish high speed digital connections; for example, VDSL2 technology can provide up to 200 Mbit/s data transfer rate (bit rate) on a single twisted pair [1]. Reaching this bit rate is achieved by such methods, which influences the physical circumstances of the links, like self-FEXT cancellation [2]. Thus, copper wire based access networks remain economical and applicable for providing telecommunications services for a long time.

Irrespectively of the material of transmission medium, access networks have numerous endpoints, and the performance of the access links is different in the case of each endpoint. Optimal operation, cost efficient renewal, maintenance, and forming correct business proposals and marketing strategies require information about the performance of network segments, moreover, in certain cases about the individual links. However, performance is unknown until the installation of the equipment of the given transmission technology, that is, until the formation of the real connections. Preliminary—and approximate—information about the performance can be gained by performance prediction techniques.

Performance prediction (sometimes also mentioned as performance evaluation) techniques are used in various fields of telecommunications. Lots of works deal with the examination of whole networks. Such methods can be found, for example, in [3–6]. Other works deal with simulation and soft computing based performance prediction, for example, [7, 8]. Besides the performance prediction of whole networks, prediction of the performance of individual links or channels

has a great importance; for example, Stupia et al. [9] or Brueninghaus et al. [10] deal with such solutions. Wireless performance prediction is made by Zhang and Ma [11] and performance predictions in symmetrical (copper) networks are mentioned by Magesacher et al. in [12]. Performance can be defined in different ways in the case of telecommunications networks. Some methods aim the prediction of the bit error ratio (BER) evolved in defined circumstances, for example, [13, 14]. Others, for example, Bosco et al. [15], predict the transmission power.

In this paper we present a novel method for predicting the available data transmission rate of the SHDSL connections to be installed over copper wire pairs of access networks, which makes the estimations from measured electrical parameters of the transmission medium by a fuzzy inference method, without the necessity of the installation of the technological equipment. This method—with small modifications—is suitable for making performance prognosis also for other transmission media and for other transmission technologies.

In the next section, those technical and physical parameters of the access networks that can be used for performance prediction are given together with our measurement setup. In Sections 3, 5.1, and 6.2 we summarize the theoretical background used in this contribution, namely, the fuzzy inferences, the wavelet analysis, and the KH fuzzy rule interpolation technique. In Section 4 we overview our previous results in performance prediction for wired telecommunications connections. In Section 6 using wavelet analyses and fuzzy interpolation technique we give a series of performance prediction algorithms based on solely the insertion loss values of the links of the access networks. At the end of the paper the results of the tests for SHDSL systems are shown, and we present a suggestion for a method efficiently applicable in industrial circumstances.

2. Physical Basics of Performance Prediction for Wired Links of Access Networks

As it can be read in Section 1, in this work the notation “performance” is used for the available maximal data transmission rate by SHDSL transmission. This value—as also other properties of the transmission—is influenced by the physical parameters of the transmission medium.

Studies were carried out for selecting those physical properties which really influence the bit rate. (The aim of the study was not the identification of lines out of order, but the performance evaluation of operable ones; thus the measurements were made on operable lines.) Performance primitives are given by the ITU-T recommendation referring to SHDSL technology [17]. Only two of them, loop attenuation defect and SNR (Signal to Noise Ratio) margin defect, are in connection with the layer 1 parameters of the line.

Loop attenuation defect occurs if the value of the loop attenuation is higher than a previously configured threshold. SHDSL loop attenuation is defined in page 43 of the recommendation as follows:

$$\begin{aligned} \text{“LA} = \frac{2}{f_{\text{sym}}} \left\{ \int_0^{f_{\text{sym}}/2} 10 \log_{10} \left[\sum_{n=0}^1 S(f - nf_{\text{sym}}) \right] df \right. \\ \left. - \int_0^{f_{\text{sym}}/2} 10 \log_{10} \left[\sum_{n=0}^1 S(f - nf_{\text{sym}}) |H(f - nf_{\text{sym}})|^2 \right] df \right\}, \end{aligned} \quad (1)$$

where f_{sym} is the symbol rate, $1/H(f)$ is the insertion loss of the loop, and $S(f)$ is the nominal transmit PSD” [17].

ITU-T also defines the SNR (in dB):

$$\begin{aligned} \text{“SNR} = \frac{1}{M} \sum_{k=1}^M 10 \log_{10} \left(1 \right. \\ \left. + \frac{S(f_{\text{sym}} - f_k) |H(f_{\text{sym}} - f_k)|^2}{N(f_{\text{sym}} - f_k)} \right. \\ \left. + \frac{S(f_k) |H(f_k)|^2}{N(f_k)} \right. \\ \left. + \frac{S(2f_{\text{sym}} - f_k) |H(2f_{\text{sym}} - f_k)|^2}{N(2f_{\text{sym}} - f_k)} \right. \\ \left. + \frac{S(f_{\text{sym}} - f_k) |H(f_{\text{sym}} - f_k)|^2}{N(f_{\text{sym}} + f_k)} \right), \end{aligned} \quad (2)$$

where $S(f)$ shall be the nominal far-end transmit signal power spectral density, $|H(f)|^2$ shall be the magnitude squared of the ideal loop insertion gain function, $N(f)$ shall be the injected crosstalk noise power spectral density, and f_{sym} shall be the transmit symbol rate” [17].

Although formulae (1) and (2) contain more variables, only two of them, insertion loss and noise, are real physical parameters of the line; thus, according to the recommendation, these two parameters of the line have influence on the available bit rate. This statement was verified by measurements on wire pairs from real telecommunication access networks in operation.

Measurements were performed as follows. Using operable, but actually unused wire pairs of copper cable based access networks, lines (similar to the subscriber loops) were temporarily created. Several electrical properties of the lines were studied by double-ended measurements. These properties were the noise, the far and near end crosstalk, insertion loss, return loss, line impedance, SNR, and the attenuation to crosstalk ratio. After measuring these electrical parameters, the central end of the line was connected to an SHDSL node and an SHDSL modem was installed to the other end of the line, and the available maximal bit rate was measured, granted a bit error ratio better than 10^{-7} . The results of the electrical property measurements were sorted into five clusters according to the measured bit rate ranges (range 1 to range 5, where the lowest bit rates belong to range 1 and the highest bit rates belong to range 5).

Statements of the recommendation, that is, only the insertion loss and the signal to noise ratio have influence on the performance, were verified by the measurements. Lots of the measured parameters, like, the return loss, do not have direct influence on the bit rate. Figure 1 depicts the measured return loss (RL) values. RL functions are coloured by the ranges of bit rates measured on related lines. It can be seen that although the colours seem to be grouped, there are no exclusive areas belonging to different bit rate ranges.

The role of the frequency dependent insertion loss in forming the data transmission speed is unequivocally verified by the results of the measurements. It can be seen in Figure 2 that curves of insertion loss cover areas belonging to individual bit rate ranges. The individual groups of colours are well separated; however, in some positions the bands of neighbouring ranges partially overlap.

Independently of the measured bit rates, the measured values of noise were similar to one another within each studied geographical area. Two examples for the results of noise measurements in two different areas (A and B) are shown in Figures 3 and 4. (Even though finding the reasons of similarity is not an aim of this study, according to the precursory examinations, it is caused probably by the fact that

telecommunications services are similar to one another in the same areas; that is, in domestic areas analog voice and ADSL transmissions are typical; thus the noise generated by them is similar in every wire pair.) Based on the results of the measurements it can be stated that even though the noise influences the available data transmission rate, this influence is the same in the case of each line in the same area. This behavior makes possible to set up hierarchical rules which are able to evaluate the performance of the wire pairs belonging to defined areas considering only the insertion loss.

Measured data were grouped according to the available bit rates. Based on the data in these groups a system of verbal rules was formulated; their general form is as follows:

$$\begin{aligned}
 R_{0-1}: & \text{ IF the form of the NOISE is similar to pattern } A \\
 & \quad \text{ THEN the performance should be evaluated by rule base } A \\
 R_{0-2}: & \text{ IF the form of the NOISE is similar to pattern } B \\
 & \quad \text{ THEN the performance should be evaluated by rule base } B \\
 & \quad \vdots \\
 R_{0-k}: & \text{ IF the form of the NOISE is similar to pattern } K \\
 & \quad \text{ THEN the performance should be evaluated by rule base } K \\
 R_{A-1}: & \text{ IF the INSERTION LOSS is huge} \\
 R_{A-2}: & \text{ IF the INSERTION LOSS is large} \\
 & \quad \text{ THEN the AVAILABLE BIT RATE belongs to range 2} \\
 & \quad \vdots \\
 R_{A-5}: & \text{ IF the INSERTION LOSS is low} \\
 & \quad \text{ THEN the AVAILABLE BIT RATE belongs to range 5} \\
 & \quad \vdots \\
 R_{K-1}: & \dots \\
 & \quad \vdots
 \end{aligned} \tag{3}$$

Subrule base ($R_A - R_K$), used in factual performance evaluation, is selected in the metalevel (R_0) of the hierarchical rule system described in (3).

Variables of the rule system (see (3)) are not numerical, but rather denote ranges. Moreover, these ranges are not clearly distinct; in many instances they partially cover each other. Evaluation of such rule systems by traditional mathematical tools is extremely difficult. Usage of exceedingly many equations may be needed for the exact mathematical presentation. Fuzzy logic introduced in the 1960s was developed for solving exactly such problems.

3. On Fuzzy Inferences

3.1. Basics of Fuzzy Set Theory. The usage of an exceptionally novel version of multivalued logics was suggested by Zadeh in

1965 [18]. The logic system introduced in [18] not only permits truth values of a statement differing from “YES” and “NO,” but extends this possibility to the infinity. The fundamental idea of Zadeh’s theoretical system is that if the phenomena of the universe can be classified into groups (sets), this classification can be numerically described and logical connections can be found between the particular phenomena, then correct and effectively handleable mathematical models of these phenomena can be produced. The operability of the system is ensured by the classifiability of the particular phenomena into special sets. The membership of the phenomena in these sets can be any value from 0 (phenomenon does not belong to the set) to 1 (phenomenon belongs absolutely to the set). Zadeh’s logical system is called “fuzzy logic” or “fuzzy set theory.” His paper published in 1965 lays the basics of this theory,

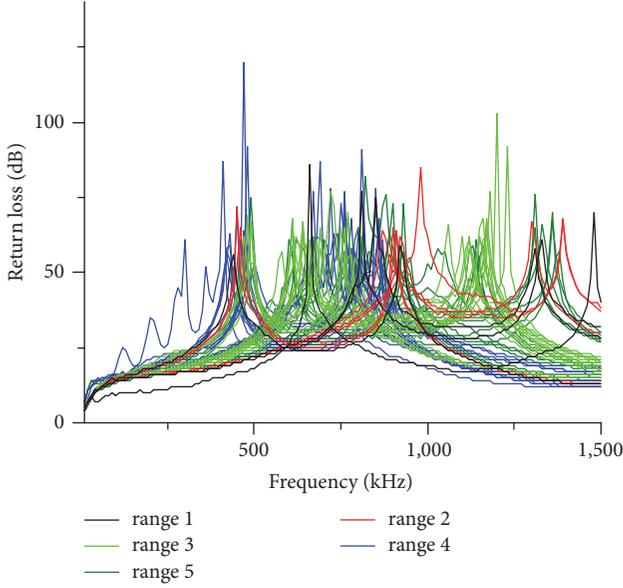


FIGURE 1: Measured return loss functions belonging to the different ranges of the bit rate.

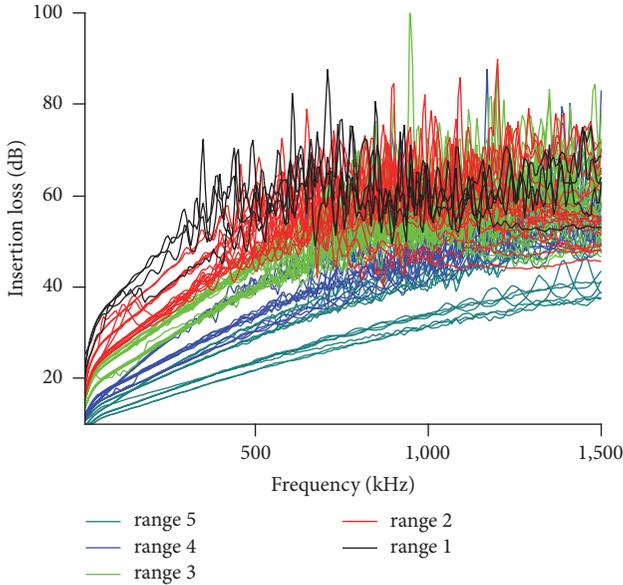


FIGURE 2: Measured insertion functions belonging to the different ranges of the bit rate.

notions of fuzzy membership value, and fuzzy membership function ($F_A(x)$)—later usually denoted by $\mu_A(x)$) is defined. As an example, Figure 5 shows a possible fuzzy set of warm temperatures.

According to [18], A' , that is, the complement of fuzzy set A , is defined as

$$f_{A'} = 1 - f_A, \quad (4)$$

where f_A is the membership function of fuzzy set A .

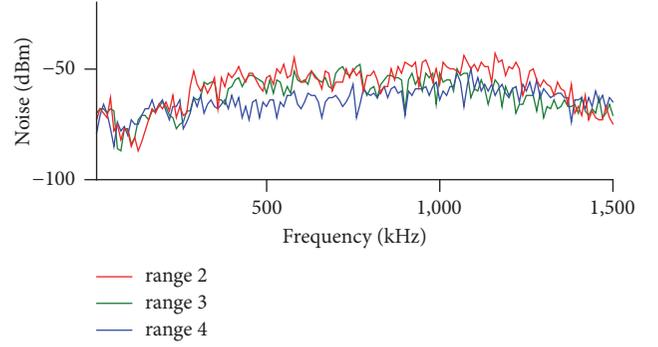


FIGURE 3: Noise of lines from area A.

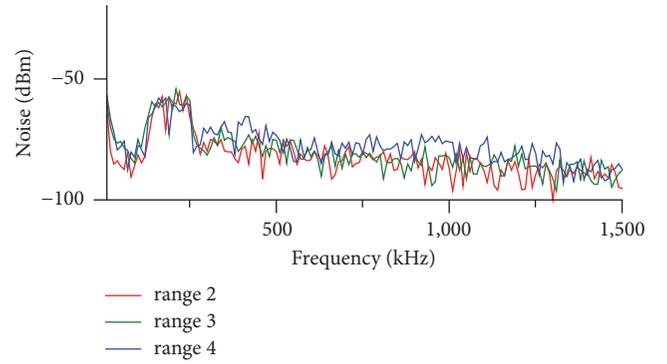


FIGURE 4: Noise of lines from area B.

The union (s -norm) of fuzzy sets A and B is as follows:

$$f_A \cup f_B = \max [f_A, f_B]. \quad (5)$$

The intersection (t -norm) of fuzzy sets A and B is as follows:

$$f_A \cap f_B = \min [f_A, f_B]. \quad (6)$$

Fuzzy norms can be implemented by numerous mathematical formulae; however, these formulae have to meet the conditions of the axiomatic systems of the different norms [19].

3.2. Zadeh's Fuzzy Reasoning. The basics of Zadeh's fuzzy set theory provide a good possibility for numerically solving such problems, which can be logically formulated. In his first publication in this topic, Zadeh denotes the methods appropriate for such solutions fuzzy algorithms [20]. Fuzzy algorithms are based on statements or commands in human language. Zadeh's examples are

$$\begin{aligned} & \text{"Set } y \text{ is approximately equal to 10 if} \\ & \quad x \text{ is approximately equal to 5"} \end{aligned} \quad (7)$$

$$\text{"If } x \text{ is large, increase } y \text{ by several units!"} \quad (8)$$

Approximately equal to 10, approximately equal to 5, large, and several are fuzzy sets, similar to the set of Figure 5. Also " x " and " y " are fuzzy sets; however, in most practical cases " x "

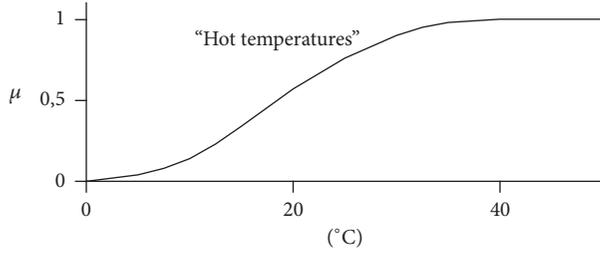


FIGURE 5: Membership values of a possible fuzzy set of hot temperatures (in °C).

is a crisp value. Using statements similar to (7) or (8), useful decision algorithms can be created, like

- “If x is small, increase y by several units!
 If x is large, decrease y by several units! (9)
 Otherwise keep y unchanged!”

[20].

At the first time, fuzzy algorithms were used for creating fuzzy inference systems in 1973 [21]. Here, fuzzy algorithms were supplemented with logic operations, arising fuzzy IF... THEN rules, moreover, fuzzy rule bases, like in

- R_1 : IF x_1 is $A_{1,1}$ AND x_2 is $A_{2,1}$ THEN y_1 is B_1
 R_2 : IF x_1 is $A_{1,2}$ AND x_2 is $A_{2,2}$ THEN y_2 is B_2
 \vdots
 R_r : IF x_1 is $A_{1,r}$ AND x_2 is $A_{2,r}$ THEN y_r is B_r . (10)

Here, (x_1, x_2, \dots, x_r) is an observation vector which can consist of fuzzy values, too, $A_{d,r}$ is a fuzzy set, where d is the number of the dimensions, r is the number of the rules, y_r is the result of the r th rule (generally also it is a fuzzy value), and B_r is the fuzzy set belonging to the output of the r th rule. The antecedent dimensions of the rules are connected to each other by AND, and the rules R_1 and R_2 by OR operator. The rule base is defined as a fuzzy relation, and the final conclusion is gained as a fuzzy composition.

“IF-THEN” fuzzy rules are used in recent control problems, too [22–24].

3.3. Mamdani’s Fuzzy Inference Method. Mamdani and Assilian suggested a more simple and practicable fuzzy reasoning method in 1975 [25]. Due to its simple implementation we also use Mamdani-like approach in this work, so its operating principles are detailed here.

Similar to Zadeh’s approach, the reasoning is based on two essential components. The first one is a rule base, and the other one is a quantified observation, which has to be evaluated by the rule base. The rule base contains the preliminary knowledge about the studied problem. It can be constructed from previous measurements of the system, or from experts’ knowledge. The possible values of the input variables are considered as fuzzy values and thus are represented by fuzzy

sets. The outputs of the rules are also fuzzy sets (outputs of the real systems are fuzzified). It has to be mentioned here that also other approaches exist; for example, Takagi and Sugeno published a fuzzy inference system in 1985, where the rules had function type outputs [26]. The other essential component is the observation itself. The observation vector contains the values of the measurements of the examined system. Theoretically, its values are considered as fuzzy values; however, in practice, most commonly they are crisp numbers.

Basic elements of the calculations are the membership values $\mu_{d,r}$ of the singular values of the observation vector \vec{x} in the fuzzy sets $A_{d,r}$ of each rule’s relevant dimension. These values are in AND logic connection with each other in the same rules, where AND corresponds to the intersection set operation. In Mamdani’s approach Zadeh’s fuzzy intersection, namely, the “minimum”, is selected to represent the logic AND. The weights w_r of the rules are calculated by

$$w_r = \min(\mu_{1,r}, \mu_{2,r}, \dots, \mu_{d,r}), \quad (11)$$

and the result of the r th rule is

$$B_r^* = \min(w_r, B_r). \quad (12)$$

The logic OR between the rules are represented by Zadeh’s s-norm; thus the final conclusion is

$$B^* = \max(B_1^*, B_2^*, \dots, B_r^*); \quad (13)$$

that is, the max–min composition of the observation vector \vec{x} and the rule base A^* .

Mamdani’s reasoning system is illustrated in Figure 6.

In most of the cases the result of a fuzzy reasoning system is also fuzzy (Takagi-Sugeno method is an exception). Although the fuzzy result can be used by another fuzzy system, for a direct application it has to be converted to a crisp value. Several methods deal with this so-called defuzzification. Two well-known ones are Center of Gravity (14) and Center of Maxima (15).

$$y_{\text{COG}} = \frac{\sum_{i=1}^r (y_i^* \cdot w_i^*)}{\sum_{i=1}^r w_i^*} \quad (14)$$

$$y_{\text{COM}} = \frac{\inf M + \sup M}{2}. \quad (15)$$

A graphical example of the differences between the results of these two techniques can be seen in Figure 7.

4. Performance Prediction for Wired Links of Access Networks

According to the hierarchical verbal rule system described by (3) and the measured values, fuzzy rule bases were constructed. The main problem in using insertion loss as antecedent is that insertion loss is a continuous function in the frequency, which does not have a closed formula. This would mean infinite possibilities for selecting antecedent dimensions, from using insertion loss values at one or more frequency points through averages to fitting a function with

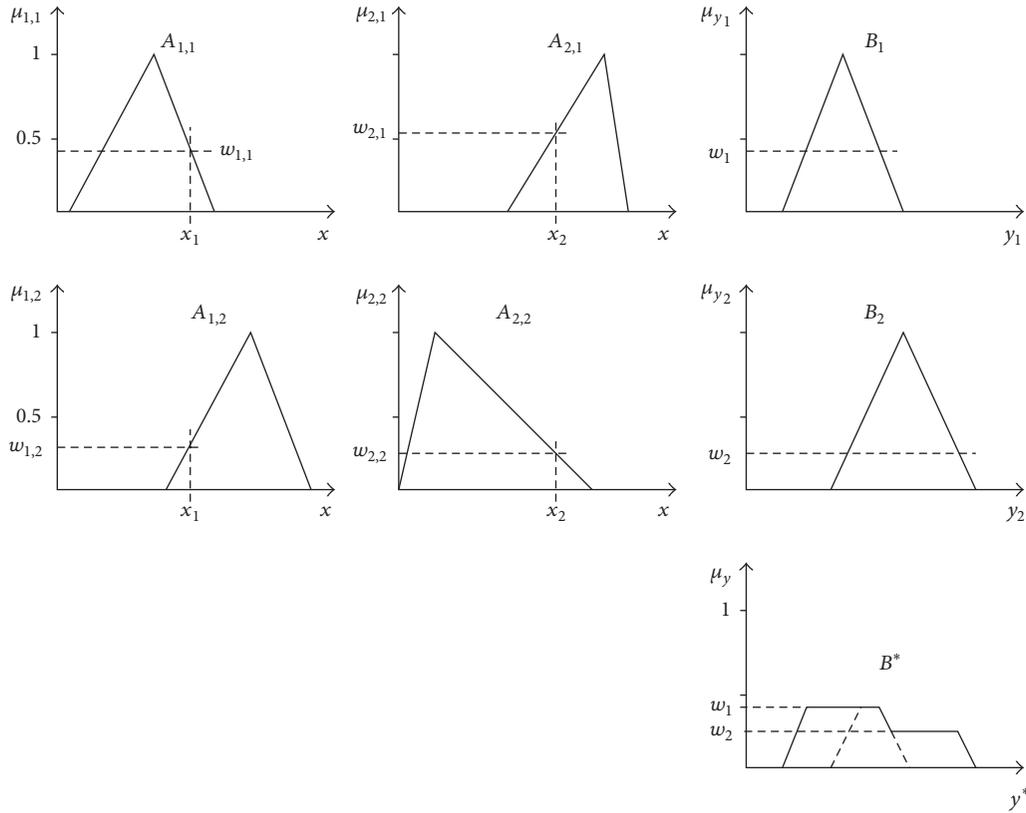
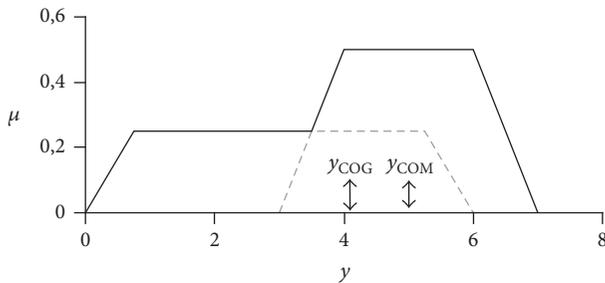


FIGURE 6: Mamdani's method.

FIGURE 7: COG and COM defuzzification of a fuzzy set ($y_{\text{COG}} = 4.016$, $y_{\text{COM}} = 5$).

some parameters to the measured values and using the fitting parameters as input vectors for the inference system.

In order to minimize the calculation, in the rule bases the characteristic points of the insertion loss were used. Based on the results of our previous research, these characteristic points in the measured 1.5 MHz wide frequency band can be at frequency points 100 kHz, 500 kHz, 750 kHz, 1000 kHz, 1250 kHz, and 1500 kHz [27]. Examinations verified that usage of more frequency points is redundant, while in case of less frequency points the effect of extreme deviations at selected frequencies can be disadvantageously dominant [28].

For the construction of the rule bases, two methods were used. In both cases the measured telecommunication access lines were separated into two parts. The training set consisted

of about half the measured lines belonging to each of the performance groups. The data of the remaining lines were used for testing.

In case of the first method, such rule bases were created in which each individual rule belonged uniquely to one of the output states. The rules had six antecedent dimensions, accordingly to the measured values of the insertion loss at the six characteristic frequencies. Fuzzy sets of these rule bases were triangular with the mean value from the training set being the α -cut belonging to $\alpha = 1$ and the minimum and maximum values of the training set determining the minimum and maximum of the support of the membership function (i.e., the $\alpha = 0$ α -cut). Its graphical example can be seen in Figure 8. Later, this rule base is referred to as RB1.

Other rule bases were created by an evolutionary algorithm [29]. The rule bases were 6 dimensional also in this case; however, the number of the rules was 10 instead of 5, and the shape of the fuzzy sets was trapezoidal. A graphical example can be seen in Figure 9.

During the evaluation Mamdani's reasoning method was used, and the fuzzy results were converted into crisp values by Center of Gravity defuzzification described by (14) [30]. The rule bases were tested using the previously selected lines from operating telecommunications access networks. In case of the lines that could be evaluated the success rate of the evaluation by each rule base was better than 94%.

During the tests, two imperfections of the evaluation system had been discovered.

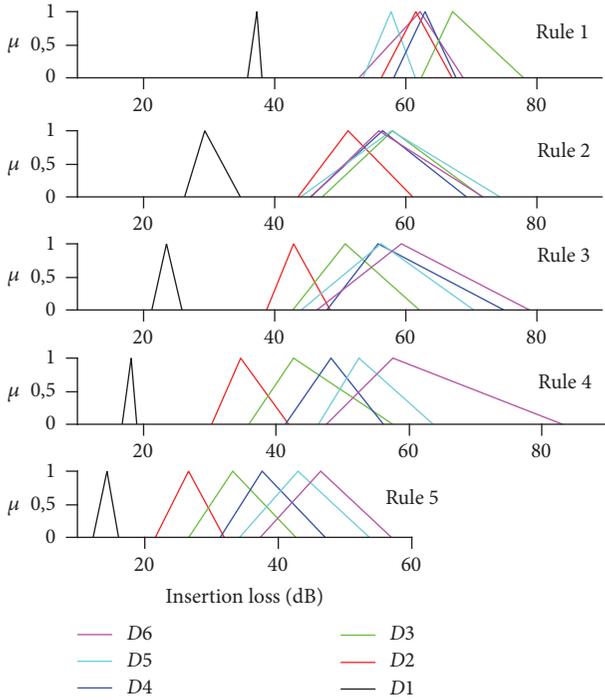


FIGURE 8: Antecedents of one of the rule bases constructed directly from measured data. Dimensions are denoted by $D1$ – $D6$.

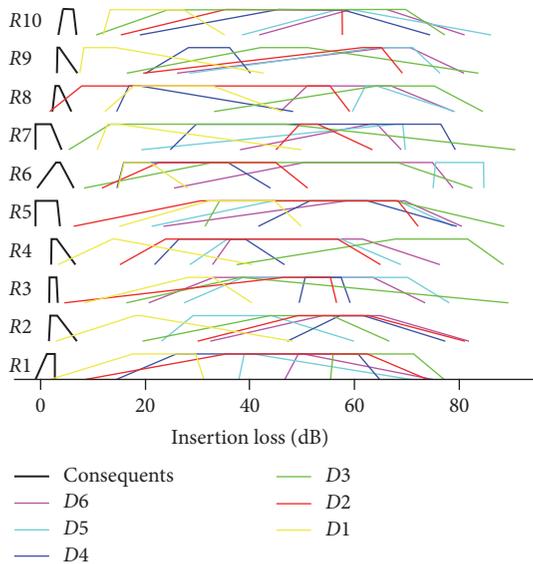


FIGURE 9: Rules 1–10 of a rule base made by evolutionary algorithm. Antecedents and consequents are trapezoidal normal (i.e., their maximum value is 1) fuzzy sets. Dimensions are denoted by $D1$ – $D6$.

One of them is originated from the sparseness of the rule bases. This problem will be referred to as vertical sparseness. As the number of the lines used in rule base construction was limited, the antecedent fuzzy sets did not cover the whole input state space; thus during the construction sparse rule bases arose. Due to the sparseness of the rule bases only those

TABLE 1: An incorrect performance prediction [16].

Rule	$D1$	$D2$	$D3$	$D4$	$D5$	$D6$	Weight
R2	0.081	0.022	0.479	0.832	0.432	0.526	0.0222
R3	0	0.848	0.854	0.852	0.223	0.791	0

lines could be successfully evaluated where the measured parameters belonged to areas covered by antecedent sets. It is important that the rule base made by evolutionary algorithm— thanks to the wider supports of fuzzy sets—was able to evaluate a higher number of lines.

The other problem will be called horizontal sparseness. As the system is based on insertion loss values at very few characteristic frequencies, extreme deviations at some of these frequencies might influence the prediction of performance but not influence the bit rates, causing this horizontal type of imperfection.

The following sections of the paper discuss the solutions for these two problems and introduce an improved approach.

5. Predictions from Insertion Loss Data of the Complete Measured Frequency Domain

As a first step let us consider the problem of horizontal sparseness.

In this case the deficiency of the method was that the performance was predicted using only some discrete values of the continuous insertion loss function, even though the whole function influences the available bit rate. Clearly the method leaves out of consideration the remaining values of the insertion loss at the other frequencies; moreover it is very sensitive to point-like deviations in the measurements. Although, thanks to the properly selected frequency points, incorrect predictions occurred in minimal number during the tests; the correct results are not guaranteed. Such an incorrect prediction can be seen in Table 1.

In case of the wire pair of Table 1, the insertion loss values at the 1st dimension and its narrow surroundings is higher than in case of the lines used in rule base construction. Moreover, even though the discrepancy is actually not high, the measured value reaches the 2nd range of the rule base. This prediction is failed, as the line, according to the bit rate measurements belongs to 3rd range; that is, the deviation of the insertion loss values at dimension 1 is so small and belongs to so narrow frequency band that the line in reality can fulfill the demands of range 3 (as it can be seen in case of the other dimensions). As we have mentioned previously, the continuous function of the insertion loss offers infinite possibilities for selecting antecedent dimensions.

Our first approach that needed technical knowledge for selecting the characteristic frequencies, but no computation from the measurement values was not successful in all the cases due to measurement uncertainty and fluctuations at the characteristic frequencies. Clearly a computationally more demanding method should be used in order to get rid of the fluctuations and take into account the insertion loss in the whole frequency domain.

5.1. Wavelet Transformations. Wavelet theory is mainly used for analyzing data, for mapping the finer and rougher structure of spatial or temporal patterns. In our case both the insertion loss and the noise are in the frequency domain; however, they have finer and rougher structure, thus naturally arising the question whether the fine structure of the data influences the performance, or it can be really neglected. It is also important to study whether an average behavior represents the insertion loss and noise better than the previously selected characteristic points. The best tool for performing both of the above tasks is to use wavelet analysis on the measured data. In order to clarify the notations, we summarize the wavelet theory in the following paragraphs. For more detailed descriptions, see, for example, [31].

Discrete wavelet analysis or multiresolution analysis (MRA) represents the functions of a Hilbert space at resolution levels. These resolution levels constitute subspaces V_j of the whole Hilbert space; each rougher subspace is embedded into the finer ones; that is, $\dots \subset V_{j-1} \subset V_j \subset V_{j+1} \subset \dots$. Each of the subspaces V_j is expanded by a set of basis functions, the so-called scaling functions ϕ_{jk} with k being the shift index and j the resolution index. All the elements of all these scaling function basis sets are generated from one mother scaling function ϕ by shrinking/extending and shifting on a regular grid as

$$\phi_{jk}(x) = 2^{j/2} \phi(2^j x - k). \quad (16)$$

Note that the grid is different at each resolution level; that is, for finer resolution the grid has smaller grid distance.

At a given resolution level a function $F(x)$ can be approximated as a linear combination of the scaling functions of that resolution level,

$$F(x) \approx F^j(x) = \sum_{-\infty}^{\infty} f_{jk} \phi_{jk}(x) \quad (17)$$

with the expansion coefficients

$$f_{jk} = \int_{-\infty}^{\infty} F(x) \phi_{jk}(x) dx. \quad (18)$$

As the subspaces V_j are embedded into one another, each function $F \in V_j$ is also element of all the finer resolution level subspaces V_{j+i} . This statement is valid also for the scaling functions, thus resulting in a refinement equation

$$\phi(x) = 2^{1/2} \sum_{i=0}^{N_s} h_i \phi(2x - i). \quad (19)$$

The constants h_i are characteristic for the various basis function systems; the number N_s gives the support of the scaling functions. For the simplest scaling function type, i.e., for the Haar scaling functions, the number of the coefficients N_c is 2, and these coefficients are $h_0 = h_1 = 1$. In the case of a more complex, $N_c = 4$ scaling function, the Daubechies-4 function is a good example, its coefficients are $h_0 = 1/\sqrt{2} \cdot (1 + \sqrt{3})$, $h_1 = 1/\sqrt{2} \cdot (3 + \sqrt{3})$, $h_2 = 1/\sqrt{2} \cdot (3 - \sqrt{3})$, and $h_3 = 1/\sqrt{2} \cdot (1 - \sqrt{3})$.

The subspaces complementing V_j in V_{j+1} are the so-called wavelet subspaces or detail spaces, and their basis functions are the wavelets. Wavelets, similarly to the scaling functions, are generated from one mother wavelet ψ as

$$\psi_{jk}(x) = 2^{j/2} \psi(2^j x - k); \quad (20)$$

moreover, they are related to the scaling functions by a formula similar to the refinement equation (19)

$$\psi(x) = 2^{1/2} \sum_{i=-N_s+1}^1 (-1)^i h_{-i+1} \phi(2x - i). \quad (21)$$

Using scaling functions as a basic resolution level approximation to a function and adding refinements to it can result in a fine resolution level approximation

$$F^j(x) = \sum_{-\infty}^{\infty} f_{0k} \phi_{0k}(x) + \sum_{i=0}^{j-1} \sum_{-\infty}^{\infty} g_{ik} \phi_{ik}(x), \quad (22)$$

where the wavelet expansion coefficients are

$$g_{ik} = \int_{-\infty}^{\infty} F(x) \psi_{jk}(x) dx. \quad (23)$$

Of course, as the fine subspace V_j is constituted from the rougher scaling function and wavelet subspaces V_{j-1} and W_{j-1} as

$$V_j = V_{j-1} \oplus W_{j-1}, \quad (24)$$

there exists a transformation, similar to the refinement equation (21) and formula (21), from the rougher level wavelets and scaling functions to the finer resolution scaling functions.

In digital data analysis we practically start from a fine resolution representation of a continuous signal (f_{jk} with a large j); this is the starting data vector (or matrix if the dimension is higher). Using the refinement equation (19) the rougher level expansion coefficients can be easily calculated from the finer resolution ones by a simple convolutional filter with filter coefficients h_i and a so-called downsampling: omitting every second element from the resulting data vector. Similar transformation produces the wavelet coefficients, with the filter constants being $(-1)^i h_{-i+1}$. Practically the transformation to the scaling function coefficients is a low-pass filtering (at about half the frequency of the highest frequency present at the starting vector), and the transformation to the wavelet coefficients is a high-pass filtering. The scaling function or low-pass coefficients describe the average behavior of the data vector, whereas the wavelet coefficients give the fine details of it.

5.2. Improved Method for Horizontal Sparseness. The problem of horizontal sparseness was solved by wavelet analysis introduced in Section 5.1. As a first step, the 1.5 MHz wide frequency range of the studied SHDSL transmission was divided into six ranges, keeping the former structure of the

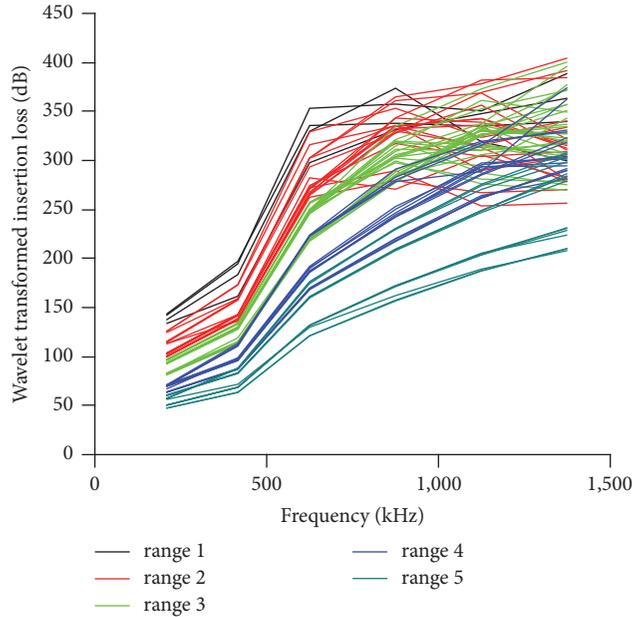


FIGURE 10: Results of the wavelet transformation of insertion loss functions shown in Figure 2.

rule bases. New rule bases were constructed using the wavelet transformed values of these six ranges. The ranges were formed as follows. The whole range was wavelet transformed into 10, than 5 values, resulting in a finer and rougher resolution representation. Due to the fact that the transmission power density is the highest in the low frequency domain, the two lowest values of the finer resolution data were used for the two lowest dimensions and the upper 4 values of the rougher resolution data for the upper four dimensions, resulting in a mixed resolution representation. Later, this type of rule bases is referred to as RBW. Figure 10 shows the wavelet transformed points. The results of the wavelet transformations are discrete values; however, for better visibility points corresponding to the same telecommunication line are connected.

According to our results, high frequency components of the wavelet transformations are not characteristic of this problem, as the available bit rate is influenced by the average heights of the insertion loss function and not by its fluctuations; thus they were not used in rule base construction. The measured insertion loss function of the line to be evaluated has to be wavelet transformed for the performance prediction. Even though this prediction seems to be made also by six discrete values, these six discrete values contain the information of the whole insertion loss function.

After doing the tests again two issues were experienced. Firstly, to achieve this result, examination of the noise was not necessary, so the hierarchical rule bases can be replaced by such a general rule base, which is constructed using wavelet transformed values of the insertion loss functions solely. Secondly, the prediction was correct in case of all wire pairs with insertion loss values belonging to the domains of the antecedent sets of the rule base. However, the problem of sparse rule bases, that is, the vertical sparseness, still existed.

6. Improvements for Handling the Sparse Rule Bases

The problem of the vertical sparseness needs other methods to overcome.

A significant degree of unsuccessful prediction is caused by the sparseness of the rule bases in case of both prediction methods introduced in Sections 4 and 5. We found two possibly efficient ways to handle this problem. One of them is the usage of fuzzy sets given by more complex membership functions, where the supports are extended to include the uncovered parts of the frequency domains in the rule antecedents. The other one is the usage of fuzzy interpolation. Both techniques were tested with the wavelet transformed functions as formerly rule bases based on wavelet transformation resulted in better predictions. The methods and the test results are described in the following.

6.1. Composite Membership Functions in Rule Antecedents. In our first method, the sparseness of the rule base in Figure 8 was canceled by using fuzzy sets described by various composite functions that had wider supports.

The first type of the composite functions consisted of four linear segments (later referred to as skirted), the second one was complete symmetric Gaussian functions (later: Gaussian), and the third one consisted of two half-Gaussian (later: semi-Gaussian) functions instead of simple triangular membership functions. The shape of these functions was formed to be similar to the shape of the original triangular antecedent sets; however, the α -cuts belonging to very small α s were much wider than the ones of the original triangular sets. All the composite rules were created from their corresponding measured and transformed minimal, maximal, and average values: IL_{\min} , IL_{\max} , and IL_{avg} .

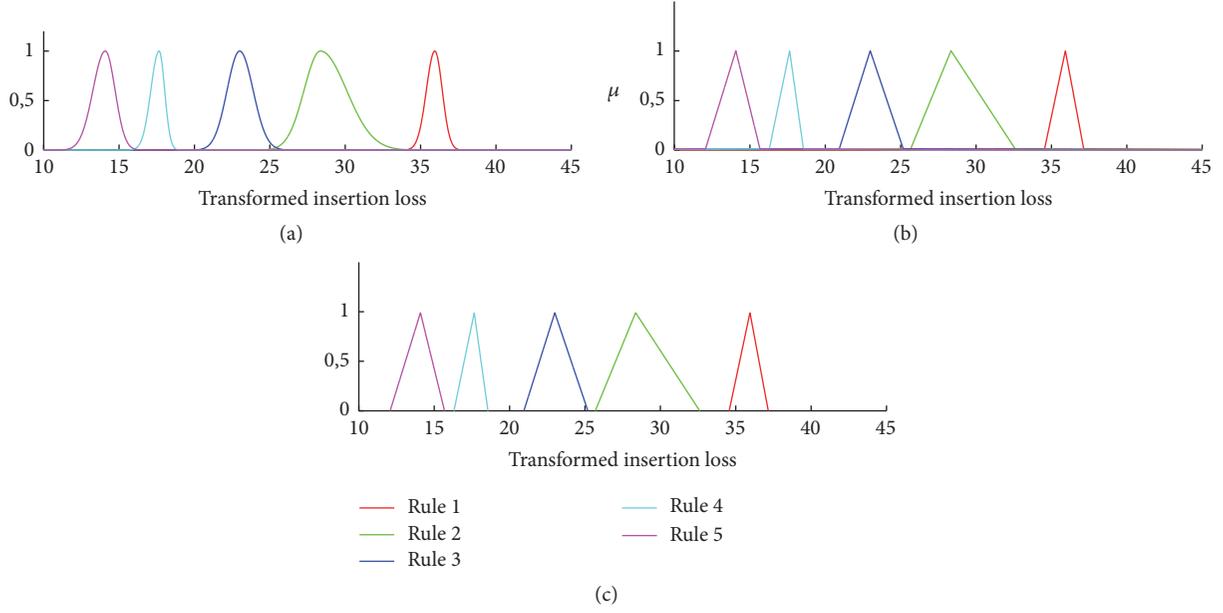


FIGURE 11: Compound membership functions of a rule. The original sparse rule can be seen in the diagram (c). Sparseness is canceled by the expanded supports of the composite membership functions of the two upper diagrams: (a) shows the half-Gaussians and (b) shows the skirted triangles.

For creating skirted triangular rules the average value was used as the core point, and the minimal and maximal values were used as the top of the skirts at $\alpha = 0.001$, whereas the upper and lower bounds of the support were set to $IL_{avg} - 10 \cdot (IL_{min} - IL_{avg})$ and $IL_{avg} + 10 \cdot (IL_{max} - IL_{avg})$.

In case of the Gaussian function formula

$$\mu(IL) = e^{-(IL - IL_{avg})^2 / 3(IL_{max} - IL_{min})^2} \quad (25)$$

was used. As Gaussian functions are symmetrical but our original measurement values based triangular rules were mostly asymmetric, for a better fitting we have introduced the functions consisting of two half-Gaussians. They were created similarly to the Gaussian one, with formula

$$\mu(IL) = \begin{cases} e^{-(IL - IL_{avg})^2 / 3(IL_{avg} - IL_{min})^2} & \text{if } IL < IL_{avg}, \\ e^{-(IL - IL_{avg})^2 / 3(IL_{max} - IL_{avg})^2} & \text{if } IL > IL_{avg}. \end{cases} \quad (26)$$

Figure 11 shows an example of a new rule constructed by half-Gaussians and skirted triangles together with the original triangular rules in Figure 11(c).

6.2. Fuzzy Interpolation. The second solution to the vertical sparseness problem was suggested by Kóczy and Hirota (KH interpolation) [33, 34]. This interpolation process creates temporary rules for the uncovered domains utilizing the nearest rule to the observation, valid only for the studied cases. More sophisticated results are provided by the stabilized KH interpolation [35]. The method creates the lower and upper bounds of the characteristic α -cuts of the corresponding fuzzy conclusion B^* by the Euclidean distance between

the infimum and supremum of the characteristic α -cuts of the antecedent sets and the conclusion. Stabilized KH interpolation is given as follows:

$$\begin{aligned} \inf \{B_{\alpha}^*\} &= \frac{\sum_{i=1}^{2n} (1/d_{\alpha L}(A^*, A_i))^k \inf \{B_{i\alpha}\}}{\sum_{i=1}^{2n} (1/d_{\alpha L}(A^*, A_i))^k}, \\ \sup \{B_{\alpha}^*\} &= \frac{\sum_{i=1}^{2n} (1/d_{\alpha U}(A^*, A_i))^k \inf \{B_{i\alpha}\}}{\sum_{i=1}^{2n} (1/d_{\alpha U}(A^*, A_i))^k}, \end{aligned} \quad (27)$$

where i denotes the number of the rules, k denotes the number of the dimensions (variables), A^* denotes the observation, A_i denotes the antecedent sets in rule i , $d_{\alpha L}(A^*, A_i)$ and $d_{\alpha U}(A^*, A_i)$ denote the lower and upper bounds of the distance between the α -cuts of observation and the antecedents, and B^* stands for the corresponding fuzzy conclusion [36].

6.3. Reduction of the Number of the Dimensions. The method described in Section 4 was based on a rule base with 6 dimensions. Its reason was that the continuous functions of physical line parameters were represented by some of their discrete values at previously selected frequencies. However, later, the information of the whole insertion loss functions was taken into consideration by the usage of the wavelet transformation method described in Section 5. Based on the results of our examinations, the reduction of the number of dimensions seemed to be possible. In order to find the optimal number of the input variables (dimensions), the insertion loss functions were wavelet transformed to different resolution levels by Haar and Daubechies wavelets, and according to these transformations, different rule bases were constructed.

TABLE 2: Techniques combined for constructing new type rule bases.

Function shape/interpolation	Number of points (Haar)	Number of points (Daubechies)
Skirted	8	4
Semi-Gaussian	4	3
Gaussian	2	2
Triangular + stabilized KH interp.	1	1

The number of the resulting points after the wavelet transformations is different in order to cover approximately the same frequency domain in the two cases.

In the calculations of the next section we have applied transformations to the starting sampled insertion loss functions down to 1, 2, 4, or 8 low-pass coefficients for the Haar basis set and to 1, 2, 3, or 4 low-pass coefficients in the case of the Daubechies 4 functions. We also kept the last high-pass coefficients (that characterize the largest scale behavior of the original function). These rule bases have the shorthand notation DB for Daubechies 4 and H for Haar wavelets and L2, L4, and so on for the number of remaining points after low-pass filtering or HLL1 if the wavelet transform resulted in one high-pass (wavelet) and one low-pass (scaling function) coefficient.

6.4. Results of the Improved Predicting Method. To create a sufficiently precise and efficient system, the methods described in Sections 6.1–6.3 were combined. Symmetrical Gaussian and asymmetrical skirted triangle and semi-Gaussian rule bases were constructed from data arising from various resolution level wavelet transformations of the insertion loss functions. Also the original triangular (sparse) rule base was improved by stabilized KH interpolation described by (27).

Another modification of the original method is that certain labels referring to the bit rate range are used in the outputs of the rules instead of consequent fuzzy sets (it is possible because each rule belongs to one and only one output state), and the final conclusion is the value belonging the label of the rule with the highest firing rate [30].

Combining the techniques from Table 2, 43 different rule bases were created. These rule bases were used in testing of 65 lines measured in five different telecommunications access networks in different circumstances (different areas: industrial, commercial, and residential at different times). The results of the tests were sorted into 3 clusters. Cluster *Correct* contained those predicted results which were equal to the measured ones. Predicted results, which ranked the evaluated line into the lower neighbour of the correct range of the bit rate, formed the cluster *Acceptable*. Other incorrect results belonged to the cluster *Incorrect*. The results classified in this way can be seen in Table 3.

According to Table 3, the best predictions were made by rule bases numbers 1 and 2. Both gave the same results for all the tested lines. The main reason for this equivalence is the very similar shapes of the fuzzy sets used in the rule bases: both of them were constructed from data of the 2-pointed wavelet transformation of the insertion loss functions using Haar wavelets. Even though the same results were gained by the first 2 rule bases, the usage of rule base number 1 is more effective, due to its easier applicability. The

method with the most accurate theoretical background, the interpolated sparse rule base constructed by Daubechies 4 wavelet transform, is still very effective. This is the rule base with 6 antecedent dimensions described in Section 5. The huge difference in the success rate between methods numbers 10 and 28 is caused by an incorrect normalization. As the whole input space is not known; thus the normalizing factor can not be correctly defined.

An interesting result is that the symmetric membership functions perform much worse than their asymmetric counterparts, which fit the measured values of the training set not the test set.

7. Conclusions

After a summary about the previously developed fuzzy inference based access network link performance prediction techniques we gave a series of improvements to the methods. The first improvement is using large-scale (averaged) insertion loss behavior, that is, wavelet transformed values instead of single measured values at characteristic frequencies, which makes the previous methods more robust to local insertion loss fluctuations.

The second improvement of the sparse rule bases is applying either membership functions with extended (widened) support (skirted triangles or a composite of two half-Gaussians) or stabilized KH fuzzy interpolation technique. Both types of methods make the previously not evaluable test lines evaluable, with correct results for almost all the testing lines. Interestingly very rough resolution level wavelet transforms of the insertion loss function (transformation down to 2 points) give the best results with rules having extended supports; however, the interpolation gives almost as good results with 6 antecedent dimensions. The worst results, that is, the highest number of not acceptable classifications, use Haar wavelets, large number of antecedent dimensions, and rules with extended support.

Recently the method is used for image classification purposes in mechanical and medical pictures.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The project was financially supported by the ÚNKP-17-4 New National Excellence Programme of Hungary and by National

TABLE 3: Results of the tests of the different rule bases.

Number	Rule base	Correct	Acceptable	Incorrect
1	Skirted – H – L2	57	8	0
2	Half-Gaussian – H – L2	57	8	0
3	RBW DB with interpolation	57	7	1
4	Skirted – H – HL1	57	6	2
5	Half-Gaussian – H – HL1	57	6	2
6	Skirted – H – L4	54	11	0
7	Half-Gaussian – H – L4	54	11	0
8	Skirted – H – L8	54	4	7
9	Half-Gaussian – H – L8	54	4	7
10	RB1 with interpolation	53	12	0
11	Skirted – DB – HL1	52	12	1
12	Skirted – DB – L2	52	12	1
13	Half-Gaussian – DB – HL1	52	12	1
14	Half-Gaussian – DB – L2	52	12	1
15	Skirted – DB – L3	51	13	1
16	Half-Gaussian – DB – L3	51	13	1
17	Interpolated – DB – L3	50	15	0
18	Skirted – DB – L1	50	14	1
19	Half-Gaussian – DB – L1	50	14	1
20	Gaussian – DB – L3	49	16	0
21	Interpolated – H – L8	49	16	0
22	Interpolated – DB – HL1	49	12	4
23	Gaussian – H – HL1	48	17	0
24	Gaussian – DB – L4	48	17	0
25	Skirted – DB – L4	48	16	1
26	Half-Gaussian – DB – L4	48	16	1
27	Interpolated – DB – L2	48	16	1
28	RB1 with normalized interpolation	46	19	0
29	RBW H with interpolation	46	19	0
30	Skirted – H – L1	45	20	0
31	Half-Gaussian – H – L1	45	20	0
32	Gaussian – DB – L2	45	20	0
33	Gaussian – H – L1	44	21	0
34	Gaussian – H – L2	44	21	0
35	Interpolated – H – L4	44	21	0
36	Gaussian – H – L4	43	19	3
37	Gaussian – H – L8	43	15	7
38	Interpolated – DB – HL1	42	19	4
39	Gaussian – DB – HL1	40	23	2
40	Gaussian – DB – L1	39	24	2
41	Interpolated – H – L2	37	28	0
42	Interpolated – H – HL1	36	29	0
43	Interpolated – H – HL1	24	41	0

In column *Rule base* the first capital (H or DB) stands for the wavelet transformation method (Haar or Daubechies), the second 1 or 2 capitals (L or HL) stand for the used components of the results of the wavelet transformations (high or high and low), and the number at the last position refers to the points resulted by the wavelet transformations.

Research, Development and Innovation Office (NKFIH), K108405 and K124055.

References

- [1] ITU: Very high speed digital subscriber line transceivers 2 (VDSL2), Technical Recommendation G.993.2, ITU Std., 2011.
- [2] ITU: Self-FEXT cancellation (vectoring) for use with VDSL2 transceivers, Technical Recommendation G.993.5, ITU Std., 2010.
- [3] D. Guo and X. Wang, "Bayesian inference of network loss characteristics with applications to TCP performance prediction," in *Proceedings of the IEEE Workshop on Statistical Signal Processing, SSP 2003*, pp. 530–533, USA, October 2003.
- [4] R. Pan, B. Prabhakar, K. Psounis, and D. Wischik, "SHRiNK: A method for scaleable performance prediction and efficient network simulation," in *Proceedings of the IEEE INFOCOM '03. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1943–1953, San Francisco, CA, USA, 2003.
- [5] B.-S. Chen, S.-C. Peng, and K.-C. Wang, "Traffic modeling, prediction, and congestion control for high-speed networks: A fuzzy AR approach," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 491–508, 2000.
- [6] L. Muka and I. Derka, "Simulation Performance Prediction in Clouds," in *Proceedings of the 9th International Symposium on Applied Informatics and Related Areas - AIS2014*, pp. 142–147, 2014.
- [7] L. Muka and I. Derka, "Evaluation and improvement of parallel discrete event simulation performance predictions: A rough-set-based approach," *Acta Polytechnica Hungarica*, vol. 13, no. 6, pp. 125–145, 2016.
- [8] G. Lencse, I. Derka, and L. Muka, "Towards the efficient simulation of telecommunication systems in heterogeneous distributed execution environments," in *Proceedings of the 36th International Conference on Telecommunications and Signal Processing, TSP '13*, pp. 304–310, Italy, July 2013.
- [9] I. Stupia, F. Giannetti, V. Lottici, and L. Vandendorpe, "A novel link performance prediction method for coded MIMO-OFDM systems," in *Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC '09*, Hungary, April 2009.
- [10] K. Brueninghaus, T. Salzer, D. Astely et al., "Link Performance Models for System Level Simulations of Broadband Radio Access Systems," in *Proceedings of the IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications '05*, pp. 2306–2311, Berlin, Germany, 2005.
- [11] J. Zhang and X. Ma, "Broadcast performance analysis of IEEE 802.11 networks under fading channels," in *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '13)*, pp. 19–21, Toronto, 2013.
- [12] T. Magesacher, D. Statovci, T. Nordström, and E. Riegler, "Performance analysis of vectored wireline systems embracing channel uncertainty," in *Proceedings of the IEEE International Conference on Communications, ICC '13*, pp. 3986–3990, Hungary, June 2013.
- [13] G. Kaddoum, F. Gagnon, P. Chargé, and D. Roviras, "A generalized BER prediction method for differential chaos shift keying system through different communication channels," *Wireless Personal Communications*, vol. 64, no. 2, pp. 425–437, 2012.
- [14] K. Sayana and J. Zhuang, "Link Performance Abstraction based on Mean Mutual Information per Bit (MMIB) of the LLR Channel," IEEE 802.16 Broadcast Wireless Access Working Group, 2007.
- [15] G. Bosco, A. Carena, R. Cigliutti, V. Curri, P. Poggiolini, and F. Forghieri, "Performance prediction for WDM PM-QPSK transmission over uncompensated links," in *Proceedings of the 2011 Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference, OFC/NFOEC 2011*, usa, March 2011.
- [16] F. Lilik, S. Nagy, and L. T. Kóczy, "Wavelet based fuzzy rule bases in pre-qualification of access networks' wire pairs," in *Proceedings of the 12th IEEE AFRICON International Conference, AFRICON 2015*, Ethiopia, September 2015.
- [17] ITU: Single-pair high-speed digital subscriber line (SHDSL) transceivers, Technical Recommendation G.993.2, ITU Std., 2003.
- [18] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [19] L. T. Kóczy and D. Tikk, "Fuzzy rendszerek (Fuzzy systems, in Hungarian)," *Typotex*, 2000.
- [20] L. A. Zadeh, "Fuzzy algorithms," *Information and Control*, vol. 12, no. 2, pp. 94–102, 1968.
- [21] L. A. Zadeh, "Outline of new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 28–44, 1973.
- [22] X.-H. Chang, J. Xiong, Z.-M. Li, and J. H. Park, "Quantized Static Output Feedback Control For Discrete-Time Systems," *IEEE Transactions on Industrial Informatics*, 2017.
- [23] X. Chang and Y. Wang, "Peak-to-Peak Filtering for Networked Nonlinear DC Motor Systems with Quantization," *IEEE Transactions on Industrial Informatics*, 2018.
- [24] X. Xie, D. Yue, H. Zhang, and C. Peng, "Control synthesis of discrete-time T-S Fuzzy systems: reducing the conservatism whilst alleviating the computational burden," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2480–2491, 2017.
- [25] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.
- [26] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [27] F. Lilik and J. Botzheim, "Fuzzy based Prequalification Methods for EoSHDSL Technology," *Acta Technica Jaurinensis*, vol. 4, no. 1, pp. 135–144, 2011.
- [28] F. Lilik, S. Nagy, and L. T. Kóczy, "Examination of Characteristic Frequencies of Insertion Loss in Performance Evaluation of Access Networks' Wire Pairs," *Acta Technica Jaurinensis*, vol. 8, no. 3, p. 267, 2015.
- [29] K. Balázs and L. T. Kóczy, "Constructing dense fuzzy systems by adaptive scheduling of optimization algorithms," *Applied and Computational Mathematics*, vol. 11, no. 1, pp. 81–101, 2012.
- [30] F. Lilik and L. T. Kóczy, "Performance evaluation of wire pairs in telecommunications networks by fuzzy and evolutionary models," in *Proceedings of the IEEE AFRICON 2013*, Mauritius, September 2013.
- [31] I. Daubechies, "Ten Lectures on Wavelets," in *CBMS-NSF regional conference series in applied mathematics 61*, SIAM, Philadelphia, 1992.
- [32] A. Haar, "Zur Theorie der orthogonalen Funktionensysteme - Erste Mitteilung," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.

- [33] L. T. Kóczy and K. Hirota, "Approximate reasoning by linear rule interpolation and general approximation," *International Journal of Approximate Reasoning*, vol. 9, no. 3, pp. 197–225, 1993.
- [34] L. T. Kóczy and K. Hirota, "Interpolative reasoning with insufficient evidence in sparse fuzzy rule bases," *Information Sciences*, vol. 71, no. 1-2, pp. 169–201, 1993.
- [35] D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon, "Stability of interpolative fuzzy KH controllers," *Fuzzy Sets and Systems*, vol. 125, no. 1, pp. 105–119, 2002.
- [36] K. Balázs and L. T. Kóczy, "Hierarchical-interpolative fuzzy system construction by genetic and bacterial memetic programming approaches," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 20, no. 2, pp. 105–131, 2012.

Research Article

Trust-Based Situation Awareness: Comparative Analysis of Agent-Based and Population-Based Modeling

Zara Nasar  and Syed Waqar Jaffry 

Punjab University College of Information Technology, Lahore, Pakistan

Correspondence should be addressed to Zara Nasar; zara.nasar@pucit.edu.pk

Received 23 December 2017; Accepted 26 March 2018; Published 8 May 2018

Academic Editor: Kevin Wong

Copyright © 2018 Zara Nasar and Syed Waqar Jaffry. 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.

In recent years, to comprehend and analyze complex systems, multiagent systems modeling and simulation are being widely used across various disciplines. Two major approaches used for multiagent systems modeling and simulation are agent-based modeling (ABM) and population-based modeling (PBM). In multiagent community, it is a silent assumption that both approaches represent similar dynamics for large population size. One of the recent studies from literature has reported similar results for a model of situation awareness spread in multiagent systems. Trust is a significant factor that affects agents' communication, and consequently it controls spread of situation awareness among agents in a multiagent system. Hence, current work firstly extends the reported model of situation awareness spread from literature, to incorporate interagent trust for both ABM and PBM. Later, these extended models are used for comparative evaluation of both approaches. Various simulation experiments for different population sizes (small and large) as well as population types (homogenous and heterogeneous) are conducted and analyzed. Results of these experiments show that for large and homogeneous population, ABM approximates behavior of PBM, but for even slightly heterogeneous population, these approaches do not produce similar results irrespective of population sizes. Thus, the current study reports that, under some conditions, ABM and PBM produce similar results for trust-based situation awareness spread in multiagent systems, but this assumption does not hold true at large.

1. Introduction

In recent years, there is an urge to comprehend complex systems in order to better understand the world. World is very complex to understand in its totality with many factors contributing towards its complexity. With the advent of sociotechnical systems, there is a dire need to analyze the relations, strengths, and weaknesses of these systems on overall environment. Thus, it is of utmost importance to understand and analyze the behavior of human beings in certain conditions, so that the emerging sociotechnical systems can serve their purpose well. Hence, practitioners and scientists are working to develop systems that can help us in understanding the underlying relations between various entities in world as well as to better comprehend and analyze the human behavior and needs. In this pursuit, multiagent systems (MAS) are on rise. MAS refer to group of multiple intelligent agents, with each having their own beliefs, desires, and intentions

and each wanting to fulfill his/her goals. These agents can communicate with each other, can influence each other's beliefs and desires, can compete and/or collaborate in order to achieve their respective goals.

The modeling and development of MAS can be carried out using variety of ways. The domain of computational modeling comprises primary concepts and constructs for modeling of real world phenomenon's using computational modeling and simulation. In context of MAS, computational modeling offers two widely used approaches, namely, agent-based modeling (ABM) and population-based modeling (PBM), respectively. ABM tends to provide individual level insights; thus, it models changes at individual level. It supports incorporation of multiple individual factors in the system, where primary actor is an agent, and systems are formed via collection of agents. The objective of ABM is to analyze the effects of individual agents having different properties on the entire system. PBM, on the other hand, as its

name suggests, focuses on providing population level/global insights regarding any phenomenon of interest. It deals with groups as whole and tends to study group dynamics at population level and their impact on the overall system. As ABM includes wide variety of parameters and offers more flexibility than that of PBM, ABM is computationally expensive than PBM and is assumed as a more realistic modeling technique. In MAS community, it is a silent assumption that if results of ABM are aggregated at population level for large population size, both ABM and PBM present similar dynamics. Thus, in this study the primary focus is to perform comparative evaluation of ABM and PBM.

In order to compare ABM and PBM, an existing study from literature is selected, which performed comparison between these two approaches as well and reported that ABM and PBM both present similar dynamics when population size is large. This existing study models the phenomenon of spread of situation awareness (SA). SA refers to flow of information regarding particular event occurrence within a group of agents. It is a perception of environment by the agent at a particular time and surrounding context followed by the projection of events in near future.

There are multiple studies that analyze and study trust dynamics in various environments and domains [1–4]. As individuals' decisions and beliefs about surroundings and environment are affected by their trust in other agents they are communicating with, trust is one factor that affects SA. Hence, trust-based SA incorporates mutual trust of communicating agents for better SA and it is being recognized as the basis of efficient group decision-making. It is of utmost importance in safety critical systems including aviation, power plant operation, and air traffic control. A brief study covering applications of the SA in general is presented in [5]. In the last couple of years, dynamics of SA are studied in variety of ways including gaming simulators, business intelligence, and online discussions using various computational techniques [6–12]. In current study, existing model of SA is being extended to trust-based SA to make the resultant model more realistic.

Hence, in this study, primary aim is to extend computational model of SA presented in [13] to trust-based SA using ABM and PBM techniques. Keeping this in view, the key goal of the current research is to analyze the proposed model with both computational modeling paradigms, namely, ABM and PBM, using homogenous as well as heterogeneous populations. In addition, comparative analysis of ABM and PBM is also presented in the light of conducted experiments. The rest of the paper describes related background, outlines methodology opted to build the system that is an extension to a previous model proposed in [13], and briefly explains the conducted experiments and respective results against various populations, followed by conclusion and future directions.

2. Background and Literature Review

Multiagent systems (MAS) are recently on rise due to advent of sociotechnical systems. MAS are being used in various domains to better comprehend and analyze the strengths,

weaknesses, and needs of human users. Some relevant studies in this regard include [14–21] that demonstrate applications of MAS in car tracking, deployment of distributed applications, land usage and land covering analysis, and serious games development regarding health, environment, and water management.

The primary research focus of the current study is comparative evaluation of the two widely used approaches for MAS modeling, namely, ABM and PBM. In this regard, there exist several studies in literature which perform comparison between these approaches. Research study presented in [22] entirely focuses on trust dynamics and performs variety of experiments using both modeling approaches followed by their comparative analysis. Another comparative study in domain of criminology is presented in [23], which studies the phenomenon of crime displacement. Several experiments with simple and complex functions are performed in this study with respect to displacement of crime. In the light of conducted experiments, study concludes that, for large population size, ABM tends to approximate the results of PBM where PBM has the advantage of being computationally efficient. Similar research study carried out in [24] is focused on epidemics and economics, which provide a comprehensive comparison on ABM and PBM approaches. It concludes that these approaches may have similar performance against certain conditions, but that is not always the case.

As far as situation awareness and trust are concerned, both these concepts are applicable only when we have multiple agents in system. Thus, in order to study relation between SA and trust, that is, trust-based SA, MAS can be used. Several studies reported in literature deal with trust-based SA. The research study carried out in [25] analyzes SA, taking interpersonal trust into account. This study models the trust-based SA using ABM. Hence, the resultant model is quite detail-oriented. In order to verify the results, air control case study was employed and results show significant effect of trust information on operation controller's SA. The concept of trust in this study is modeled using various factors that include domain knowledge of trustee, interpersonal trust, trust in information source, and trust of an agent in its own self. The resultant model is quite complex and, consequently, cannot be modeled using PBM with such minor attention to details. Other research studies include [26, 27] that study the relation between trust and SA in the context of Diner's Dilemma game and autonomous vehicles, respectively.

As the existing model for trust-based SA [25] cannot be readily converted into its population-based counterpart, the current study extends a relatively simple model of SA spread as presented in [13] with incorporation of interagent trust using both ABM and PBM. This existing study models the spread of SA within the group using both ABM and PBM approaches. Later, comparative analysis of both modeling approaches is also presented. In the light of conducted experiments, it concludes that both modeling approaches represent similar dynamics if the population size is large.

Therefore, in the current study, extended model of trust-based SA is being used in order to perform comparative evaluation of these modeling approaches. Moreover, experiments are performed using homogenous as well as heterogeneous

populations using various values for interagent trust. Current study is an extension of our previous work [28], which is majorly focused on model formulation for trust-based SA and it performs experiments using homogenous population only.

3. Methodology

In order to perform the comparative evaluation, first of all, the existing model presented in [13] is extended to incorporate interagent trust. This section briefly covers the details regarding existing model and its proposed extension. Case study used is similar to that of original model, which belongs to Air Traffic Management domain. Case study is about a couple of flights, where some are in line and waiting for signal/command to take off. At a time, only one air vehicle is signaled for takeoff. The case focuses on a scenario when a pilot misinterprets the command and starts to take off, whereas another vehicle was already signaled for takeoff on crossing runway at the same time. Both vehicle crews were unaware that another vehicle is also on the runway. Control tower, after analyzing the situation, signaled one of the flight pilots to abort the takeoff. Pilot, in this situation, had to check whether it is possible to abort the takeoff by applying brakes or not. Eventually, after analyzing the situation, the pilot instructed copilot to stop takeoff by applying brakes. During braking, crew saw other flight flying close in with distance of five meters. Thus, a very serious accident was prevented. Further details regarding case study are presented in [29].

In the context of above study, there exist three belief states of agents and populations that include correct, incorrect, and unknown. For the sake of simplicity, model carries only one interesting phenomenon against whom these belief states are being set. This assumption is being held in original study as well. The aim is to analyze the dynamics of overall system and transition of agents/populations from one state to another over the passage of time, when interagent trust is also taken into account.

3.1. Existing Model. This section briefly covers the nomenclature and formulations of the existing model that is being followed in proposed model as well.

3.1.1. Global Properties. The existing model made use of the following six global state transition probabilities (STPs) for ABM:

- (1) $\mathbf{p(c, i)}$: STP from correct to incorrect belief state,
- (2) $\mathbf{p(c, u)}$: STP from correct to unknown belief state,
- (3) $\mathbf{p(u, i)}$: STP from unknown to incorrect belief state,
- (4) $\mathbf{p(u, c)}$: STP from unknown to correct belief state,
- (5) $\mathbf{p(i, c)}$: STP from incorrect to correct belief state,
- (6) $\mathbf{p(i, u)}$: STP from incorrect to unknown belief state.

In the existing study, the PBM is modeled on the grounds of widely used epidemic spread model for population where each subpopulation is represented separately and their inter-transition is modeled on the basis of transition rates. In

order to map the ABM and PBM model, differences of respective properties in ABM model are used as state-changing parameters in PBM model. Gamma γ represents the transition rate and U, C, and I refer to size of population carrying unknown, correct, and incorrect belief states. Thus, transition rates are modeled using following equations:

$$\begin{aligned}\gamma(U, C) &= p(u, c) - p(c, u) \\ \gamma(U, I) &= p(u, i) - p(i, u) \\ \gamma(I, C) &= p(i, c) - p(c, i).\end{aligned}\tag{1}$$

3.1.2. State Transition Rules. In ABM, rules to determine belief state of an agent at next time point are dependent on following inputs:

- (1) agent's own belief state at time-step \mathbf{t} ,
- (2) state transition probabilities (STPs mentioned in Section 3.1.1),
- (3) belief state of communicating agent.

Whenever two agents communicate, a random number is drawn; if its value is less than or equal to respective inter-STP of belief states, belief state of receiver agent is updated. Differential equations for PBM to compute population counts against every belief state depend on the following parameters:

- (1) rate of interstate transitions,
- (2) population count at current time-step, against every belief state.

The respective differential equations for population-based model are as follows:

$$\begin{aligned}\frac{dC}{dt} &= U(t) * C(t) * \gamma(U, C) + I(t) * C(t) * \gamma(I, C) \\ \frac{dI}{dt} &= U(t) * I(t) * \gamma(U, I) - C(t) * I(t) * \gamma(I, C) \\ \frac{dU}{dt} &= -C(t) * U(t) * \gamma(U, C) - I(t) * U(t) \\ &\quad * \gamma(U, I).\end{aligned}\tag{2}$$

In (2), $\mathbf{C(t)}$, $\mathbf{I(t)}$, and $\mathbf{U(t)}$ represent counts of subpopulations at time-step \mathbf{t} carrying correct, incorrect, and unknown belief state, respectively.

3.2. Proposed Model. Trust is an individual factor and an agent can have different trust values for all other agents in a given environment. Trust value 0.5 is regarded as the mean trust value of an agent. Trust value greater than 0.5 shows higher trust and vice versa, where trust can reside in the range of $[0, 1]$ and is modeled as nonnegative number.

In order to map the existing model, global properties that are used in existing ABM model are made local to every agent. Thus, every agent has total of six transition probabilities along with trust vector, carrying respective agent's trust in all other agents. This setup tends to provide flexibility for generating

```

/* Both Agent A and Agent B carry correct belief state */
if Belief_State(A, t) is correct and Belief_State(B, t) is correct
then Belief_State(A, t + 1) is correct
/* Agent A and Agent B carry correct and incorrect belief states respectively */
if Belief_State(A, t) is correct and Belief_State(B, t) is incorrect and  $r \leq (2 * \text{trust}(A, B) * p(c, i))$ 
then Belief_State(A, t + 1) is incorrect /* Agent A adopts the belief state of Agent B */
if Belief_State(A, t) is correct and Belief_State(B, t) is incorrect and  $r > (2 * \text{trust}(A, B) * p(c, i))$ 
then Belief_State(A, t + 1) is correct /* Agent A does not adopt the belief state of Agent B */
/* Agent A and Agent B carry correct and unknown belief states respectively */
if Belief_State(A, t) is correct and Belief_State(B, t) is unknown and  $r \leq (2 * \text{trust}(A, B) * p(c, u))$ 
then Belief_State(A, t + 1) is unknown /* Agent A adopts the belief state of Agent B */
if Belief_State(A, t) is correct and Belief_State(B, t) is unknown and  $r > (2 * \text{trust}(A, B) * p(c, u))$ 
then Belief_State(A, t + 1) is correct /* Agent A does not adopt the belief state of Agent B */

```

PSEUDOCODE 1

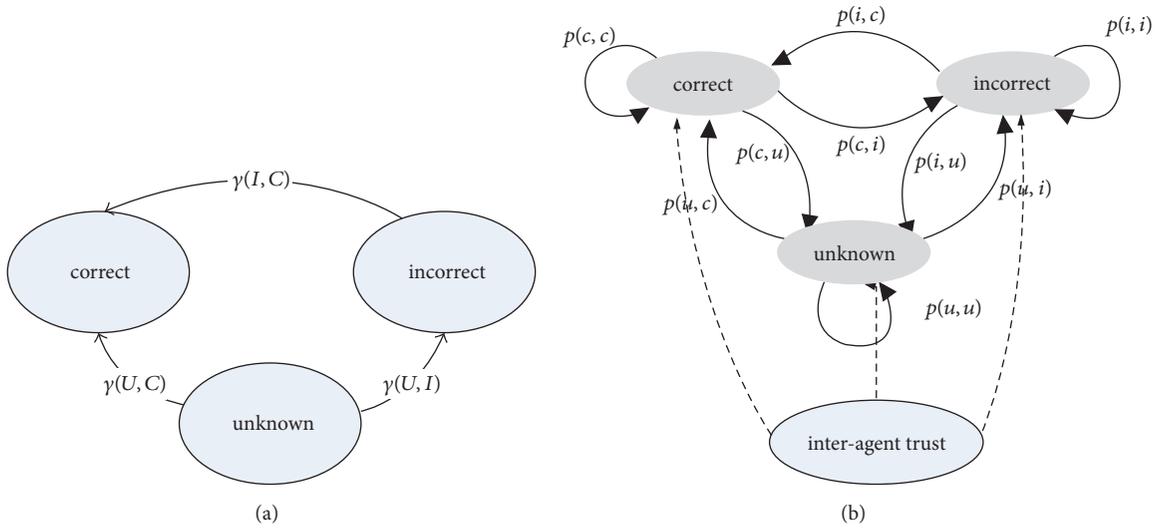


FIGURE 1: Interstate transition models for ABM and PBM. (a) State transition model for PBM; (b) state transition model for ABM partially adopted from [13].

variety of homogeneous and heterogeneous populations by means of interagent trust, transition probabilities, and combination of these two.

For PBM, in order to calculate transition rate (Gamma- γ), all individual transition probabilities against agents are averaged, keeping overall community trust in account. Figure 1 represents overall transitions and flow involved in proposed model. Figures 1(a) and 1(b) present the state transitions involved in PBM and ABM, respectively. Figure 1(b) extends the original model figure as presented in [13], where connections made via dashed lines represent the extensions in the proposed model. This section explains the interstate transition for ABM and PBM in extended model.

3.2.1. State Transition Rules for ABM. Transition model for agent-based communication is dependent on the belief states of communicating entities along with their trust in each other. Following pseudocode presents the scenario when agent A with correct belief state (receiver) is receiving information from agent B (sender). Similar code is being used when

an agent with incorrect or unknown belief state is on receiver end. The selection of sender agent is performed via randomly selecting any agent from the overall set of agents, as proposed in existing model [2]. When an agent A (receiver) communicates with agent B (sender), state transition is governed by Pseudocode 1, where **Belief_State(A, t)** returns the belief state of agent A at time-step **t**. **trust(A, B)** returns the trust of agent A on agent B, and **p(x, y)** represents STPs as described in global properties of existing model. **r** is a uniform random number that is drawn from range [0, 1]. In order to keep model comparable with existing mode, factor of 2 is being multiplied with interagent trust and inter-STP. This computational model thus results in the formation of original model when interagent trust between two agents is 0.5, by making the effect of trust neutral and only using the respective inter-STP.

3.2.2. State Transition Rules for PBM. The transition equations for PBM are the same as that of the existing model. In context of PBM, difference lies in calculation of global

TABLE 1: Parameters and population settings for nominal and wishful-thinking scenarios.

Scenario	Parameters for ABM					
	$p(u, c)$	$p(u, i)$	$p(c, u)$	$p(c, i)$	$p(i, u)$	$p(i, c)$
<i>Nominal</i>	0.20	0.15	0.10	0.02	0.05	0.04
<i>Wishful-thinking</i>	0.15	0.10	0.10	0.02	0.05	0.01
	Parameters for PBM			Initial population distributions		
	$\gamma(U, C)$	$\gamma(U, I)$	$\gamma(I, C)$	Unknown	Correct	Incorrect
<i>Nominal</i>	0.10	0.10	0.20	50%	25%	25%
<i>Wishful-thinking</i>	0.05	0.05	-0.01	50%	~38%	~12%

transition probabilities only, which are computed by means of averaging transition probabilities against every agent keeping its interagent trust into account.

$$Global_p(U, C) = \frac{\sum_{k=1}^N P(u, c)_i}{N} * 2 \quad (3)$$

* *mean_trust*

$$Global_p(U, I) = \frac{\sum_{k=1}^N P(u, i)_i}{N} * 2 \quad (4)$$

* *mean_trust*

$$Global_p(C, U) = \frac{\sum_{k=1}^N P(c, u)_i}{N} * 2 \quad (5)$$

* *mean_trust*

$$Global_p(C, I) = \frac{\sum_{k=1}^N P(c, i)_i}{N} * 2 \quad (6)$$

* *mean_trust*

$$Global_p(I, C) = \frac{\sum_{k=1}^N P(i, c)_i}{N} * 2 \quad (7)$$

* *mean_trust*

$$Global_p(I, U) = \frac{\sum_{k=1}^N P(i, u)_i}{N} * 2 \quad (8)$$

* *mean_trust*

$$mean_trust(agenti) = \frac{\sum_{j=1}^N trust(x_i, x_j)}{N} \quad (9)$$

$$mean_trust = \frac{\sum_{i=1}^N mean_trust(agenti)}{N} \quad (10)$$

In (3)–(8), **Global_p(X, Y)** refers to the STP from some belief state X to another belief state Y , at population level. N in this equation represents the total number of agents/population size. **mean_trust(agent i)** represents average trust that i th agent puts in the rest of community. **mean_trust**, on the other hand, represents the average trust in overall community. $p(x, y)_i$ refers to the inter-STP, against i th agent, from some belief state x to belief state y , that is, local to agent i . After calculating population level probabilities using

(3)–(8), the following set of equations are used in order to calculate state transition rate in PBM:

$$\begin{aligned} \gamma(U, C) &= Global_p(U, C) - Global_p(C, U) \\ \gamma(U, I) &= Global_p(U, I) - Global_p(I, U) \\ \gamma(I, C) &= Global_p(I, C) - Global_p(C, I). \end{aligned} \quad (11)$$

In the original model, the two scenarios “nominal” and “wishful-thinking” are presented. Nominal scenario refers to the situation where mental states of agents and their respective expectations from the environment are not biased. Wishful-thinking or group thinking refers to the situation where agents tend to incorporate incorrect beliefs in order to fulfill their desires, as incorrect beliefs appear to be more wishful and desirable. Existing study carries default values of parameters against these scenarios. Default parameters and initial population settings that are used in existing model are presented in Table 1.

4. Experiments

Experiments conducted can be broadly classified into homogeneous and heterogeneous categories. In the context of current research study, the effect of interagent trust on spread of information is focused on; thus, the interagent trust values act as basis for categorization.

Homogeneous population refers to population where participating agents carry similar configurations; that is, they are alike. In current study, in case of homogenous population, every agent carries similar trust value for the rest of agents; that is, trust of overall community is similar to trust of an individual agent in any other. The original model can be reproduced in extended model by means of homogenous population where all agents carry mean trust value of 0.5 in each other.

Heterogeneous environments, on the other hand, deal with cases when agents in environment differ from each other; that is, agents are unlike. In extended model, heterogeneity can be involved at various levels. It can be modeled with respect to trust or inter-STPs or both. Major challenge in heterogeneous population is to deal with the variety of possibilities. For example, consider granularity of trust to be 0.1: this means that total ten possibilities can exist for trust assignment. Now as we have total of three subpopulations of different beliefs, if each subset of population is to be experimented with various assignments of trust, total possibilities

TABLE 2: Transition probabilities for PBM against different values of trust in both scenarios.

Trust	Nominal scenario			Wishful-thinking scenario		
	$\gamma(U, C)$	$\gamma(U, I)$	$\gamma(I, C)$	$\gamma(U, C)$	$\gamma(U, I)$	$\gamma(I, C)$
0.1	0.02	0.02	0.004	0.01	0.01	-0.020
0.3	0.06	0.06	0.012	0.03	0.03	-0.060
0.5	0.10	0.10	0.020	0.05	0.05	-0.010
0.7	0.14	0.14	0.028	0.07	0.07	-0.014
0.9	0.18	0.18	0.036	0.09	0.09	-0.018

will be $10^3 = 1000$. Consider the granularity of trust to be 0.01: in this case total possibilities for experiments become $100^3 = 1000,000$. Thus, exhaustively performing all possible set of experiments is not feasible due to huge set of possibilities. In other words, the level of abstraction is of huge importance in context of heterogeneous experiments.

Currently, the experiments that are conducted on heterogeneous population follow the same transition equations and model as proposed in Section 3.2. There exist multiple options to further introduce heterogeneity by means of using distributions rather than concrete trust values or by updating trust after agents communicate with each other. Current study focuses on assignment of trust values, where trust can either be low, medium, or high, as means of heterogeneity.

The rest of this section focuses on homogeneous and heterogeneous experiments that are carried out using extended model. All the experiments in this study are carried out using MATLAB.

4.1. Homogeneous Experiments. Experiments are carried out following similar experimental setup to that described in original model. The parameter settings described in Table 1 are employed in order to perform the experiments. Original study tends to experiment with two population sizes: eight and hundred, whereas simulation values are recorded up to 250 time-steps. In addition, to avoid randomness, ABM results are averaged over 1000 simulations for every experiment performed. In the original model, the results against eight agents (small population size) do not result in similar behavior against both computational modeling paradigms, whereas in case of hundred agents, both techniques generate similar behavior, as affirmed in many studies in relevant literature.

As the proposed model carries additional feature of interagent trust, simulation values are recorded up to 500 time-steps in order to check for convergence of results for ABM and PBM. Transition probabilities for both scenarios, which are computed via proposed approach for PBM, are presented in Table 2. To get an overall idea and for the sake of brevity, results are being reported on lower trust value of 0.1 and higher trust value of 0.9 against both scenarios. This section presents the experiment's results against homogenous population, where each color-coded graph represents time-steps along x -axis and population size/number of agents along y -axis. In addition, each graph carries ABM results with solid line and PBM results with dashed line.

The proposed approach models trust as nonnegative entity where mean trust value is 0.5. Trust value greater than 0.5 represents high trust and vice versa. With mean trust value of 0.5, the original model can be produced using the proposed formulation. Figure 2 shows the results generated via original model and the proposed model. Results show that existing model is among the subset of models that can be produced via presented approach.

In case of both nominal and wishful-thinking scenarios, for smaller population size, ABM and PBM do not approach each other, whereas, for larger population sizes, the dynamics of ABM and PBM represent similar behavior. Thus, the hypothesis carried from original study regarding dynamics of ABM and PBM with respect to population sizes is verified. Figure 3 presents the results of both computational techniques against nominal scenario with population sizes of eight and hundred, along with trust value of 0.1 and 0.9. Figure 4 focuses on the results achieved, when proposed model is used in wishful-thinking scenario.

One thing to note in homogenous population results is that PBM trend remains almost consistent in small population size as well as large population size. ABM dynamics, on the other hand, gets heavily affected by the total population size. The comparative study shows that for large population sizes, ABM tends to approximate the results of PBM, where PBM is very efficient and its dynamics are independent of the population size. On the contrary, computational complexity of ABM increases with the number of agents. Thus, in order to model small population size, PBM cannot be used if individual's biases are to be taken into account.

In addition to these insights, results also show that if trust community has high trust factor, agents having unknown belief state tend to adopt the correct/incorrect belief states, depending on the scenarios and transition rates, quite quickly. On the other hand, in case of low trust, the transition from unknown to correct/incorrect state takes relatively longer time. The primary hypothesis regarding this study about relation with SA and trust is thus confirmed.

4.2. Heterogeneous Experiments. Heterogeneous experiments are also conducted on the settings presented in Table 1. In order to conduct the experiments, trust is categorized into three groups, low, medium, and high, with values of 0.1, 0.5, and 0.9, respectively. The reason for hard-coding the value is to have ease in analysis of results. In the current set of conducted experiments, a trust value being

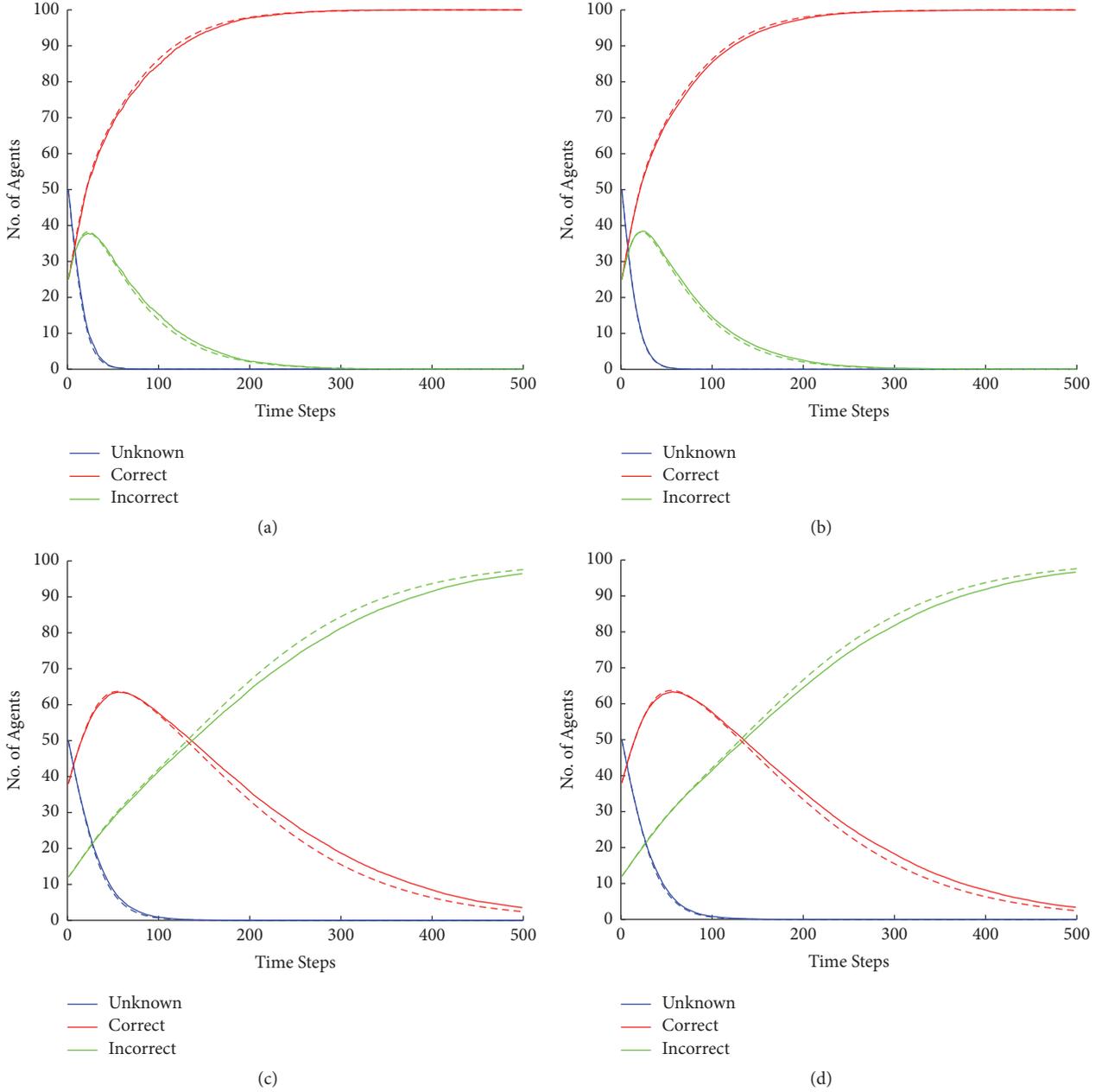


FIGURE 2: Comparison of original and proposed model with mean trust value of 0.5 and 100 population size. (a) Nominal scenario in existing model; (b) nominal scenario in proposed model; (c) wishful-thinking scenario in existing model; (d) wishful-thinking scenario in proposed model.

either low, medium, or high is assigned to subpopulation carrying a particular belief.

Total of twenty-seven configurations are possible using current experiment settings. After conducting all experiments against both scenarios, resulting graphs were firstly analyzed visually in order to look for patterns. Later averaged root mean square error (RMSE) was calculated against all belief states using temporal simulation traces of PBM and ABM for every experiment using (12). In these equations, N represents total number of agents/population size. The overall trend of RMSE in both

scenarios and values against all configurations are provided in Appendix A.

$$\text{RMSE}_{\text{Correct}} = \sqrt{\frac{\sum_{i=1}^N (\text{PBM}_{\text{correct}_i} - \text{ABM}_{\text{correct}_i})^2}{N}}$$

$$\begin{aligned} \text{RMSE}_{\text{Incorrect}} \\ = \sqrt{\frac{\sum_{i=1}^N (\text{PBM}_{\text{Incorrect}_i} - \text{ABM}_{\text{Incorrect}_i})^2}{N}} \end{aligned}$$

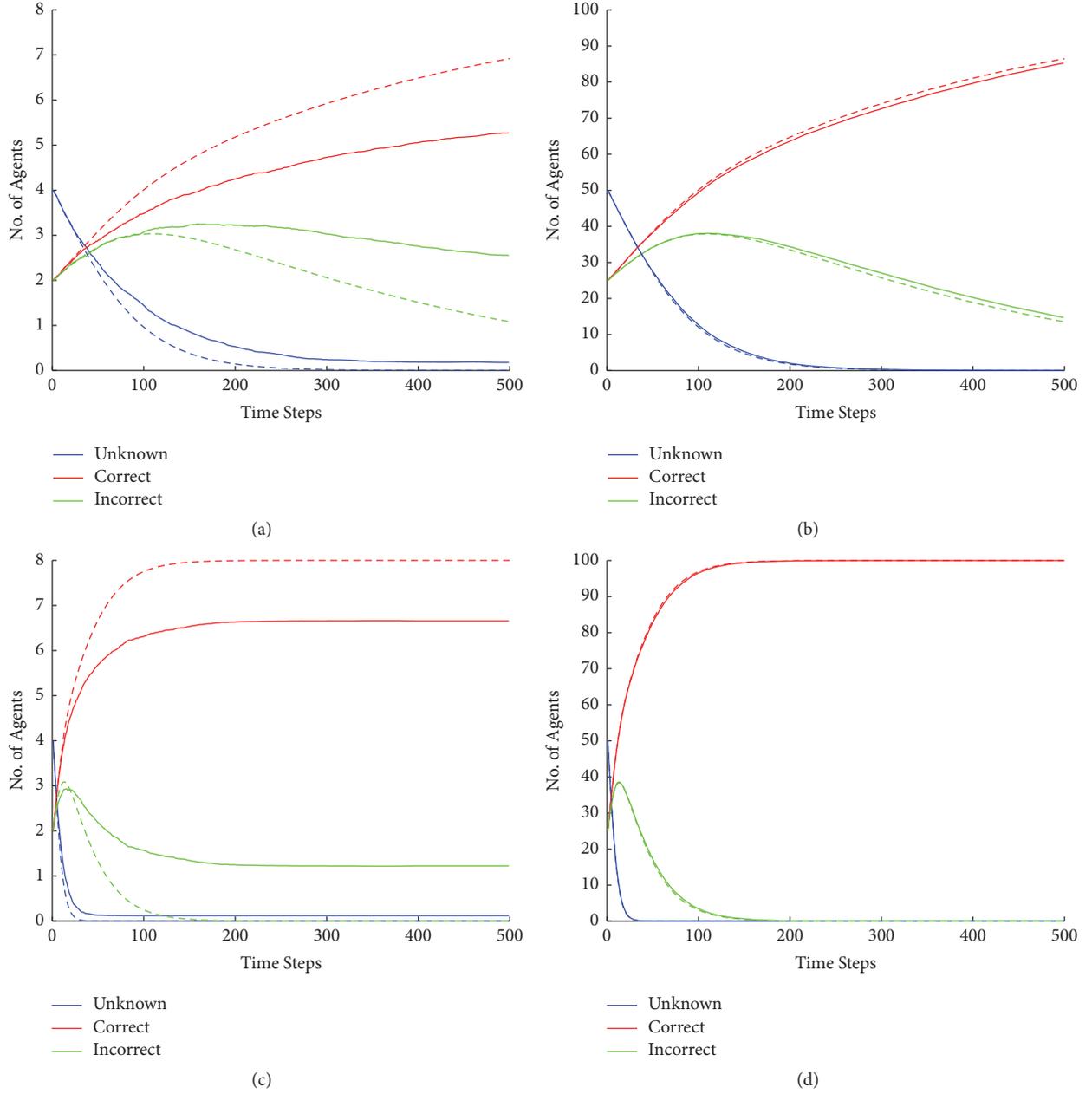


FIGURE 3: Dynamics in “nominal” scenario against low and high trust values with different population sizes. (a) Trust = 0.1, population size = 8; (b) trust = 0.1, population size = 100; (c) trust = 0.9, population size = 8; (d) trust = 0.9, population size = 100.

$$\begin{aligned}
 & \text{RMSE}_{\text{Unknown}} \\
 &= \sqrt{\frac{\sum_{i=1}^N (\text{PBM}_{\text{Unknown}_i} - \text{ABM}_{\text{Unknown}_i})^2}{N}} \\
 & \text{Average}_{\text{RMSE}} \\
 &= \frac{\text{RMSE}_{\text{Correct}} + \text{RMSE}_{\text{Incorrect}} + \text{RMSE}_{\text{Unknown}}}{3}.
 \end{aligned} \tag{12}$$

As affirmed in homogenous experiments, both ABM and PBM almost produce similar trends over the time. As total

of these 27 configurations consist of three homogenous populations as well, the RMSE among homogenous populations was minimum compared to the rest. Thus, it further validates our experiment assumption that ABM and PBM show similar trends in homogenous populations.

An interesting finding in the light of visually analyzing results and by means of error function is that “almost homogeneous populations” also perform well in both scenarios. One intuition for this can be the similarity between these populations and homogenous populations. As these configurations are very close to homogenous configurations, respective results are better than the rest as shown in Figure 5.

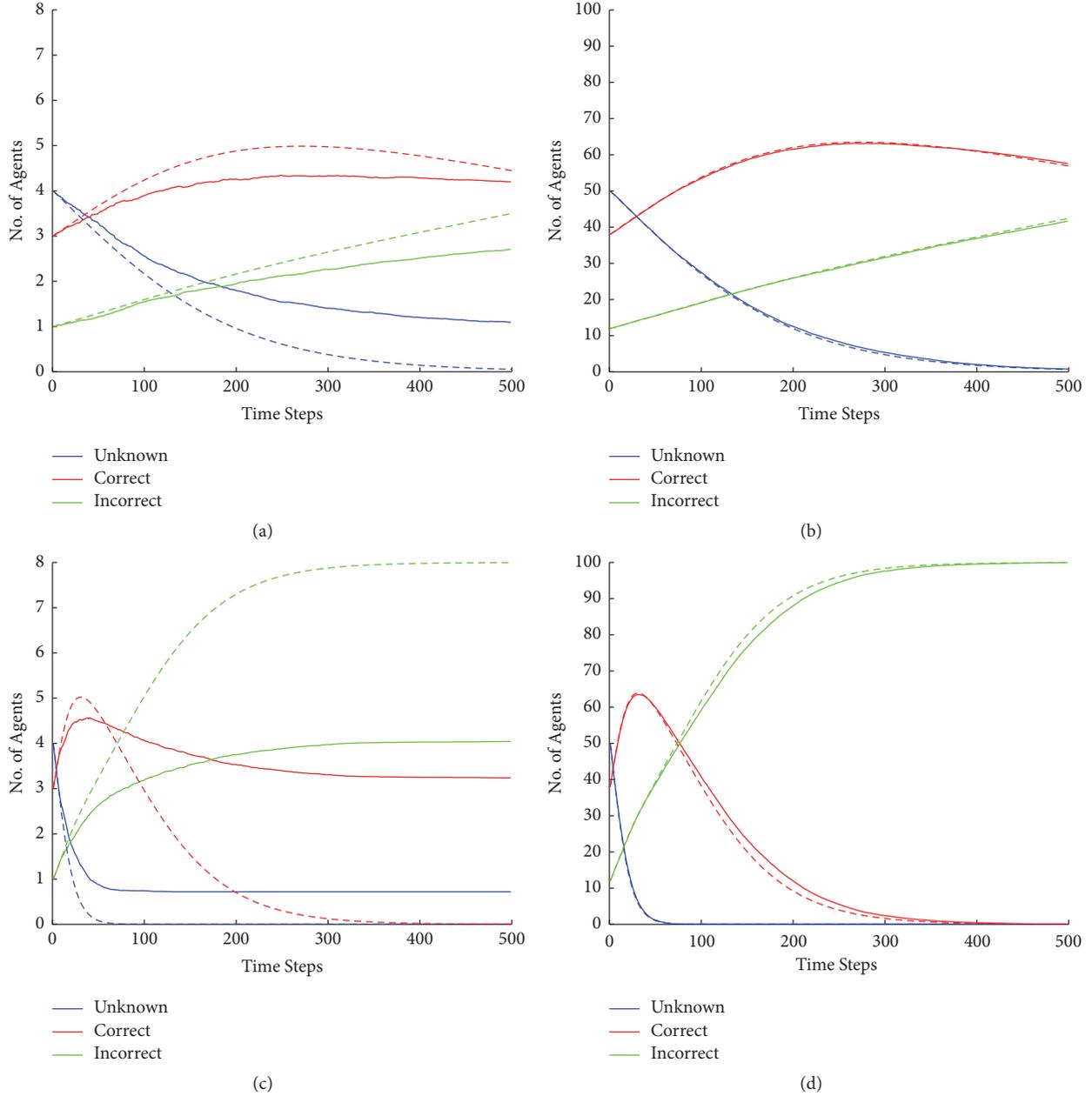


FIGURE 4: Dynamics in “wishful-thinking scenario” against low and high trust values with different population sizes. (a) Trust = 0.1, $N = 8$; (b) trust = 0.1, $N = 100$; (c) trust = 0.9, $N = 8$; (d) trust = 0.9, $N = 100$.

These cases are best performers after homogenous populations. Details of overall RMSE results against twenty-seven scenarios up to 500 time-steps are shared in Appendix A. Figure 5(a) presents the system where trust configurations are HMM; that is, unknown community has high trust whereas correct and incorrect communities have medium trust in the overall community. The configuration is coded in the order unknown, correct and incorrect. Similarly, Figure 5(b) presents the scenario when unknown and correct communities place high trust whereas incorrect community places medium trust in the overall population.

The only exception to the above observation is when unknown community has low trust in the rest. If the net transitions in the model are analyzed, it gets clear that unknown community ends up opting either incorrect or correct belief states as represented in Figure 1(a). Now, if unknown community has low trust in the rest, in case of ABM, the resulting transitions would be quite slow in comparison to PBM. Due to this slow convergence in ABM, the difference between ABM and PBM is increased, though eventually they either converge to each other or make an asymptote. Figure 6 presents the scenarios where unknown

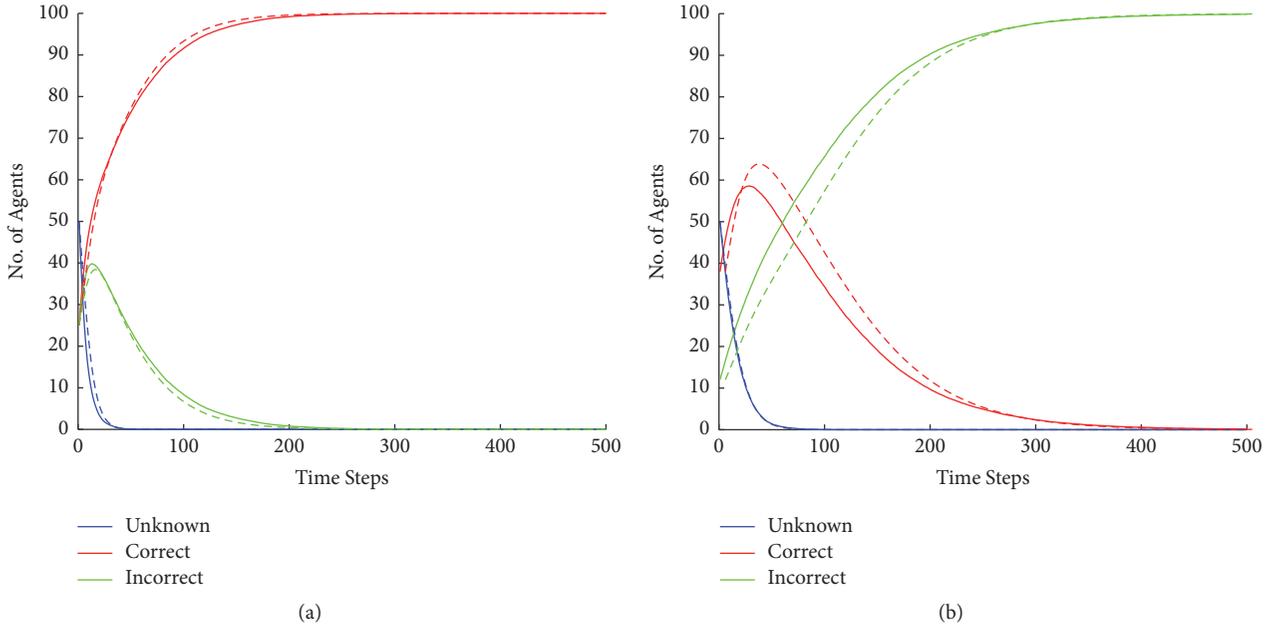


FIGURE 5: Results against almost homogenous populations for both scenarios. (a) HMM case against “nominal scenario”; (b) HMM case against “wishful-thinking scenario.”

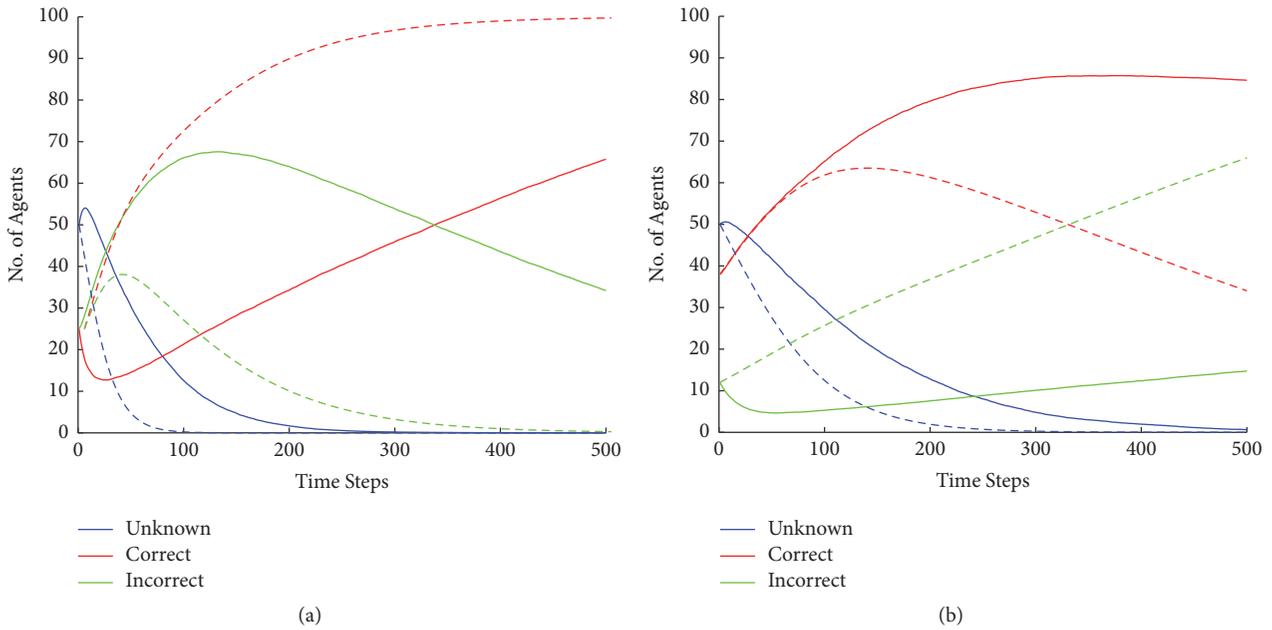


FIGURE 6: Results against both scenarios where unknown population has low trust in the community. (a) LHL case against “nominal scenario”; (b) LLH case against “wishful-thinking scenario.”

community has low trust in others. Population configurations presented in Figure 6 include LHL in case of nominal scenario that carries the highest RMSE in comparison to other configurations; for wishful-thinking scenario, LLH case is included, that is, the third most erroneous configurations in terms of RMSE.

If both scenarios are analyzed closely, it is evident that nominal scenario favors correct belief spread and eventually

majority of population will end up opting correct belief whereas wishful-thinking scenario favors incorrect belief and eventually population will carry incorrect belief. These final states can be regarded as sink. Next pattern that is observed in conducted experiments highlights the effect of low trust on sink community. Hypothesis in this regard is that as the eventual state to converge is the same as that of sink and additionally sink community has low trust in others,

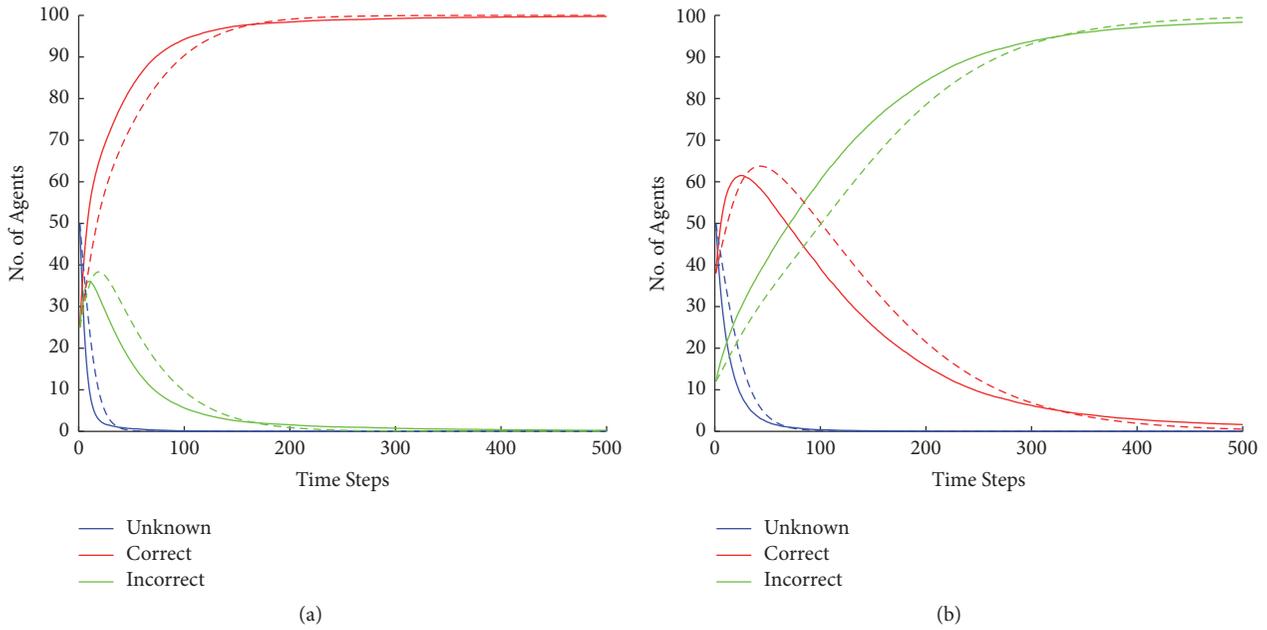


FIGURE 7: Results against both scenarios where SINK has low trust in the community. (a) HLM case against “nominal scenario”; (b) HLM case against “wishful-thinking scenario.”

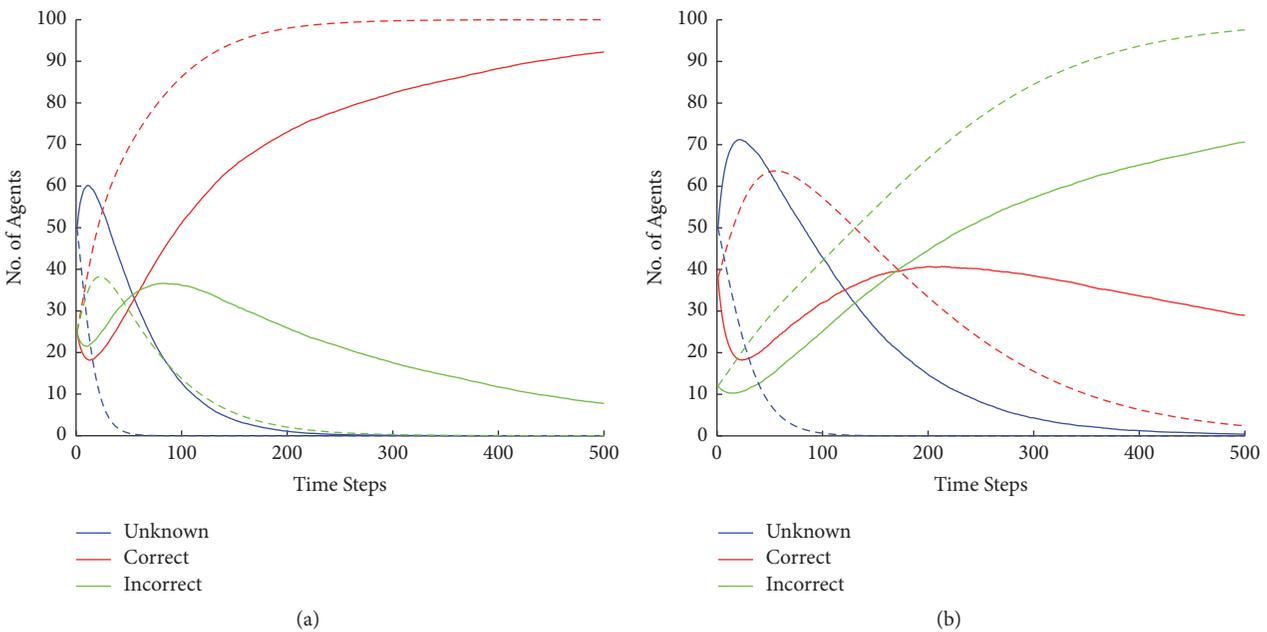


FIGURE 8: Results against both scenarios against similar configuration of trust. (a) LHH case against “nominal scenario”; (b) LHH case against “wishful-thinking scenario.”

transition from sink to others would be very slow. Thus, earlier convergence will happen in case of ABM than in that of PBM, and overall differences would be less. Consequently, results would be better matched and overall error will be reduced. Figure 7 presents the results when sink has low trust, where correct and incorrect belief states are sinks in nominal and wishful-thinking scenarios, respectively.

Figure 8 presents the trends against both scenarios in a similar case. As both scenarios have different sinks, the configuration selected for comparison is the one with similar trust for both correct and incorrect populations. In case of nominal scenario, ABM and PBM tend to approach each other earlier than that of wishful-thinking scenario. One hypothesis in this regard is that wishful-thinking scenario

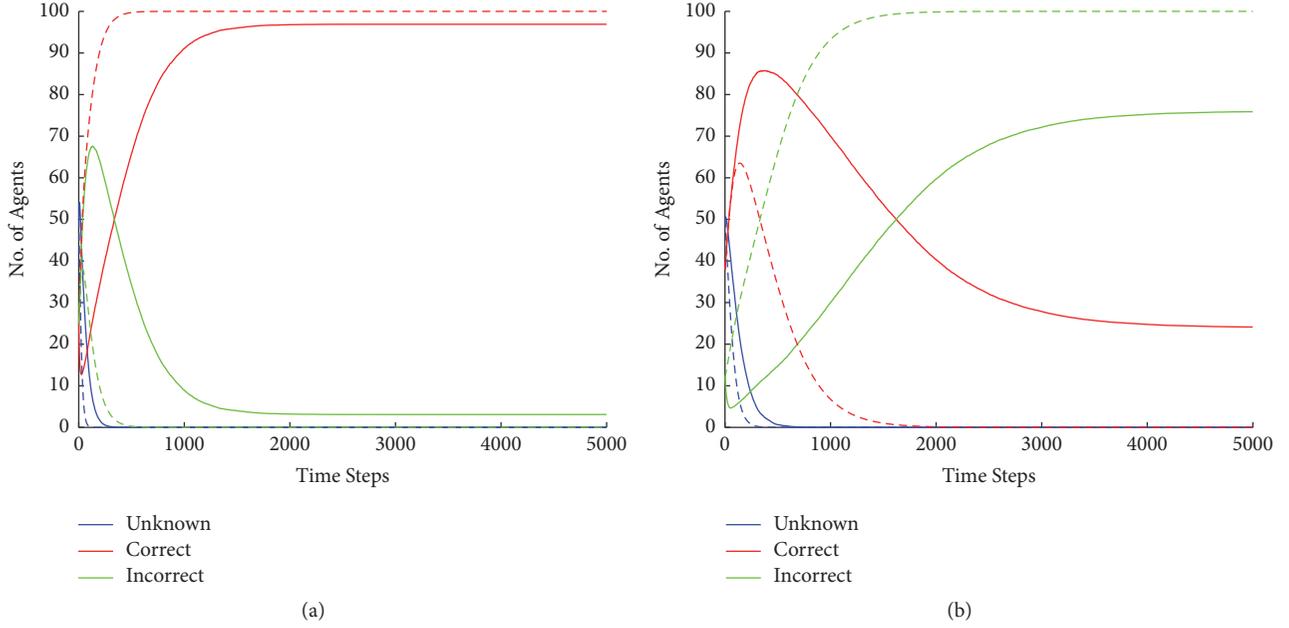


FIGURE 9: Asymptotes formation in both scenarios. (a) LHL case against nominal scenario; (b) LLH case against wishful-thinking scenario.

has lower inter-STP than nominal scenario as presented in Table 1.

Another offshoot of low inter-STPs is that, in some particular cases, where unknown community puts low trust in others, not all population eventually transit towards sink. This is because, due to lower interstate STP, the transition rate among states already gets low. In addition, as the major community that is to be transited to other communities (i.e., the unknown population) also has lower trust in population, resulting transition rate would be very low. Due to their resultant affect, the whole population does not end up following the same belief. Thus, asymptotes are formed as represented in Figure 9. Here, in this particular experiment, simulation values are recorded up to 5000 time-steps in order to present the dynamics of the overall system more clearly and precisely.

In the light of the above points, one can clearly analyze the power of ABM in contrast to PBM. Both ABM and PBM tend to model the effect of inter-STPs as reported in Appendix B. Apart from that, PBM tends to model various configurations in a similar way, whereas ABM gets affected by the individual agent's biases towards the system in terms of both transition rate and trust as reported in Appendix A. Thus, ABM has the capability to model the dynamics of system taking into account agents' local properties and biases. PBM, on the other hand, tends to model phenomenon at quite abstract level. Which one of them is really true in real world can only be determined by empirical validation.

5. Conclusion and Future Work

In this study, comparison of two widely used approaches to model multiagent systems (MAS) is performed, namely,

agent-based modeling (ABM) and population-based modeling (PBM). ABM is assumed to be more realistic and insightful as it has capability to model individual level characteristics, but due to this very reason, it is computationally expensive. PBM is usually employed to analyze global insights, and thus it is computationally efficient as it is independent of population size. Regarding these two modeling approaches, there is a prevailing assumption in MAS community that both approaches present similar dynamics for large population. Therefore, it is of interest to determine whether or not these two modeling approaches present similar dynamics in case of large population size.

In order to perform the comparative study, trust-based situation awareness was modeled by means of extending an existing study from literature. This existing study also performs comparative evaluation of these approaches and concludes that both approaches present similar dynamics if population size is large. This study models the spread of situation awareness (SA) within a group of agents. As trust is among primary factors that can affect SA spread, current study extends this model to incorporate interagent trust, thus proposing formulation for trust-based SA through both ABM and PBM approaches.

To compare ABM and PBM, experiments are conducted using homogenous and heterogeneous populations with respect to interagent trust factor. Results show that, in case of homogenous population, both approaches present similar dynamics when population size is large as affirmed in existing literature studies.

Heterogeneous experiments' dynamics, on the other hand, are sensitive to initial trust assignments as well as community wide trust distributions. Thus, dynamics exhibited by both approaches in heterogeneous populations tend to vary with respect to parameter configurations. In various

TABLE 3: RMSE against all configurations for both scenarios.

Unknown	Trust distribution		Nominal scenario		Wishful-thinking scenario	
	Correct	Incorrect	Mean trust	RMSE	Mean trust	RMSE
L	L	L	0.1	0.8688	0.1	0.5521
L	L	M	0.2	6.5583	0.148	16.3115
L	L	H	0.3	7.3172	0.196	24.518
L	M	L	0.2	22.2928	0.252	16.7516
L	M	M	0.3	13.2705	0.3	16.3012
L	M	H	0.4	12.278	0.348	24.0071
L	H	L	0.3	33.7182	0.404	17.2442
L	H	M	0.4	20.8549	0.452	18.6414
L	H	H	0.5	18.6118	0.5	24.3485
M	L	L	0.3	6.0913	0.3	9.6563
M	L	M	0.4	4.4084	0.348	19.854
M	L	H	0.5	4.3772	0.396	25.752
M	M	L	0.4	10.1467	0.452	6.7352
M	M	M	0.5	0.5007	0.5	1.5256
M	M	H	0.6	1.6807	0.548	6.5738
M	H	L	0.5	13.0625	0.604	8.0953
M	H	M	0.6	3.5929	0.652	3.7707
M	H	H	0.7	1.9148	0.7	3.9007
H	L	L	0.5	8.1663	0.5	16.032
H	L	M	0.6	3.2983	0.548	23.3535
H	L	H	0.7	3.9246	0.596	27.1409
H	M	L	0.6	10.6268	0.652	4.5747
H	M	M	0.7	1.0245	0.7	2.7398
H	M	H	0.8	1.4448	0.748	5.7717
H	H	L	0.7	12.0641	0.804	6.0571
H	H	M	0.8	2.2877	0.852	2.2345
H	H	H	0.9	0.3569	0.9	1.0679

scenarios, both approaches exhibit dynamics that are different in the start of simulation but eventually converge to each other. Moreover, some of the input configurations result in equilibrium state with respect to the number of agents against each modeling approach, but both approaches do not converge to each other. Thus, it could be concluded that, for trust-based SA, these approaches do not behave alike when population is heterogeneous.

Therefore, this study deserves further empirical work to validate which one of these approaches performs better in reality for trust-based SA models. If it is empirically validated that ABM's dynamics are closer to reality, then community should focus on the efficient implementation of ABM models. On the other hand, if PBM's results are found to be more realistic, then community should focus on developing and extending existing PBM models which carry additional advantage of being computationally efficient as well. After determination of empirically validated modeling approach, the respective model can be extended to incorporate further related concepts such as group hierarchy in subpopulations. Additionally, state transition rules can also be updated in order to model relatively more realism for trust-based SA.

Appendix

A. Root Mean Square Errors

This section is focused on results of root mean square error (RMSE) against heterogeneous populations. RMSE is used in order to compare the results of ABM and PBM against similar trust distributions. As total of twenty-seven configurations are possible, Table 3 lists the RMSE against every configuration for both scenarios along with mean trust. Mean trust represents the average trust of the whole community against every configuration that is primarily used in the formulation of PBM dynamics. Table entries are made in the chronological order with respect to trust configuration.

By closely analyzing the trend of RMSE in Table 3, it is quite evident that for various trust distributions mean trust remains the same; for example, the configurations LHH, MMM, and HLL all result in mean trust value of 0.5 with different RMSEs against both scenarios. Dynamics for configurations that result in mean trust value of 0.5 against both scenarios are presented in Figure 10 (entries in boldface in Table 3). Each color-coded figure represents dynamics

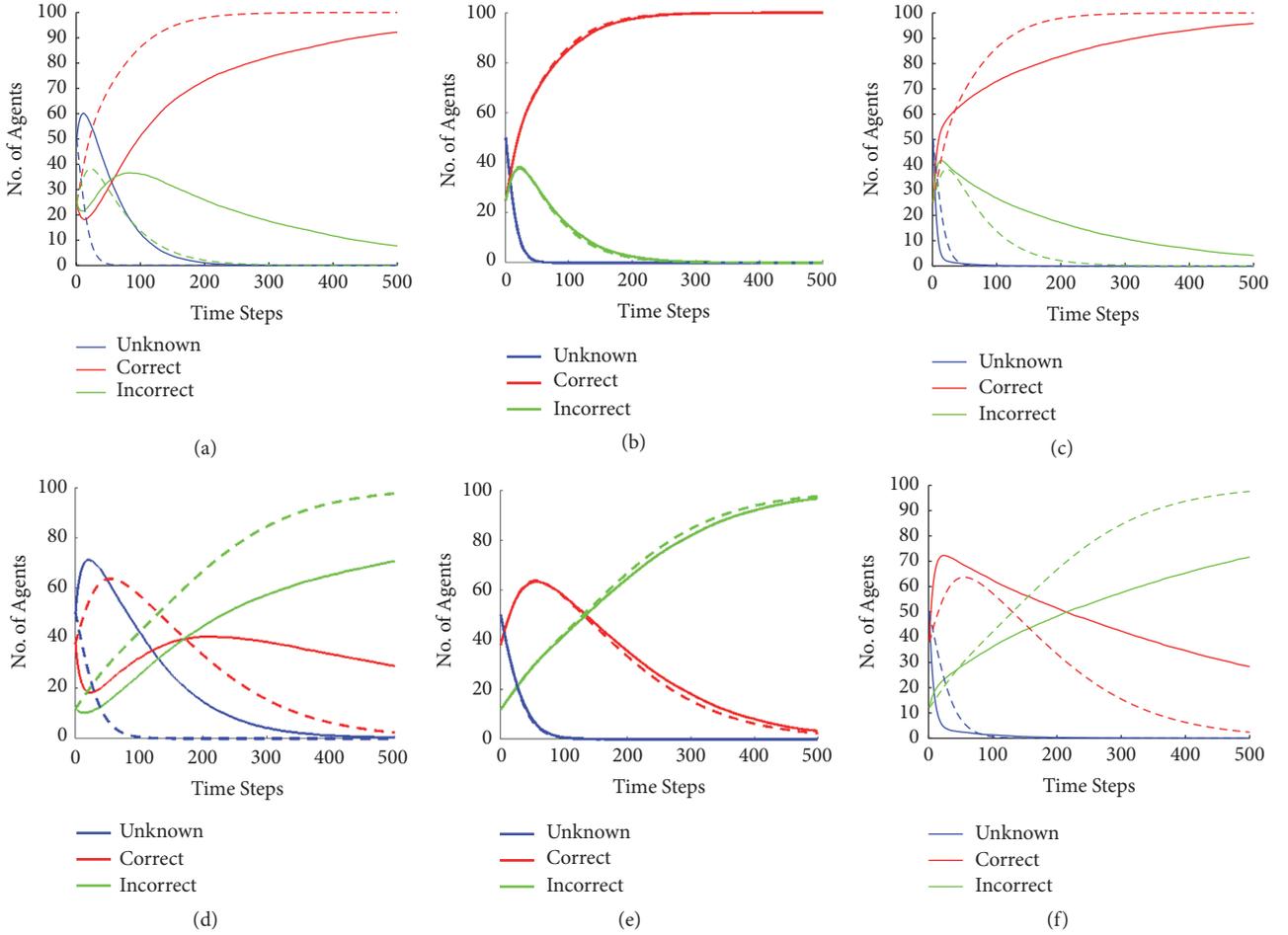


FIGURE 10: Dynamics in various configurations that result in similar mean trust against PBM against both scenarios. (a) LHH case in nominal; (b) MMM case in nominal; (c) HLL case in nominal; (d) LHH case in wishful-thinking; (e) MMM in wishful-thinking; (f) HLL in wishful-thinking.

exhibited by PBM with dashed lines whereas results against ABM are represented with solid lines.

As it is evident in Figure 10, various population configurations exhibit similar behavior in case of PBM, whereas trends vary in case of ABM depending upon the population distribution. This is because unlike PBM, ABM has the ability to incorporate and eventually project the dynamics of system, keeping agents' properties and biases in account.

B. Effect of Transition Rates

State transition rate probabilities (STPs) play a significant role in terms of convergence. The higher the STP is, the faster the convergence will happen. In order to analyze the effect of transition rate in the proposed model, an experiment was carried out. In this experiment, traces are generated against nominal and wishful-thinking scenarios with their default transition rates as well as with the updated transition rates reduced by 50%. Figure 11 represents the trends when normal and reduced transition rates are employed against MLM configuration, that is, a configuration where unknown

population have medium trust, correct population have low trust, and incorrect population have medium trust in the rest of the community. Figures 11(a) and 11(b) represent the trends in nominal scenario when original and reduced STPs are employed whereas Figures 11(c) and 11(d) show the trends in wishful-thinking scenario with original and reduced transition rates. Each color-coded figure represents dynamics exhibited by PBM with dashed lines whereas results against ABM are represented with solid lines.

If we analyze Figures 11(a) and 11(b), it is evident that dynamics in (a) tends to converge after approximately 250 time-steps, whereas dynamics exhibited in (b) show convergence after 500 time-steps approximately. Thus, both systems represent similar dynamics, with the primary difference of respective time required. Similar behavior is presented in Figures 11(c) and 11(d). In these figures, simulations are recorded up to 5000 time-steps in order to provide clear idea about the overall dynamics of the system. If we analyze the trend of PBM curve for correct population (one in red and represented via dashed line) in both figures, it is clear that in (c) respective curve gets to zero after approximately 1000

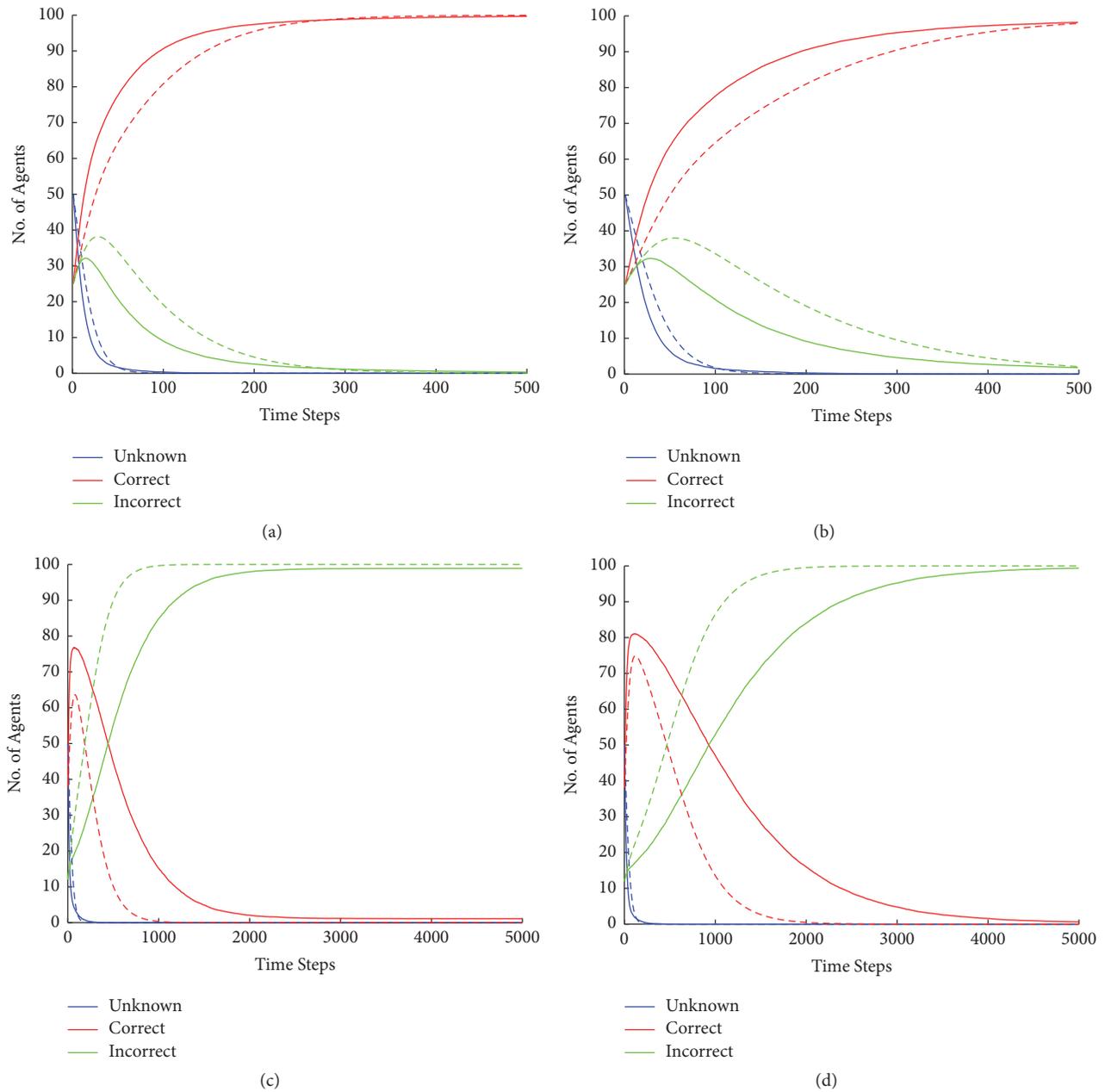


FIGURE 11: Results against both scenarios against reduced transition rates. (a) MLM case against "nominal scenario" with original transition rates; (b) MLM case against "nominal scenario" with reduced transition rates; (c) MLM case against "wishful-thinking scenario"; (d) MLM case against "wishful-thinking scenario" with reduced transition rates.

time-steps, whereas in (d) it takes almost double the time, that is, approximately 2000 time-steps to get a similar result. Thus, in the light of the above points, we can say that transition rate affects the overall convergence in the context of both ABM and PBM.

Disclosure

Part of this paper has been presented at the International Conference on Advancements in Computational Sciences

(ICACS-2018) [28]. The current paper extends the conference paper by providing additional simulation experiments in heterogeneous population, more detailed analysis of simulation results, and a more extensive discussion regarding the behavior of agent-based and population-based modeling in various environments.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] R. Falcone and C. Castelfranchi, "Trust dynamics: how trust is influenced by direct experiences and by trust itself," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004*, pp. 740–747, New York, NY, USA, July 2004.
- [2] M. Hoogendoorn, S. W. Jaffry, and J. Treur, "Modeling dynamics of relative trust of competitive information agents," in *Proceedings of the Cooperative Information Agents XII*, pp. 55–70, Prague, Czech Republic, 2008.
- [3] M. Hoogendoorn, S. W. Jaffry, and J. Treur, "An adaptive agent model estimating human trust in information sources," in *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2009*, pp. 458–465, Milan, Italy, September 2009.
- [4] S. W. Jaffry and J. Treur, "Modelling trust for communicating agents: Agent-based and population-based perspectives," in *International Conference on Computational Collective Intelligence*, vol. 6922 of *Lecture Notes in Computer Science*, pp. 366–377, Springer, Berlin, Germany, 2011.
- [5] N. A. Stanton, P. R. G. Chambers, and J. Piggott, "Situational awareness and safety," *Safety Science*, vol. 39, no. 3, pp. 189–204, 2001.
- [6] T. Bosse, R.-J. Merk, and J. Treur, "Modelling temporal aspects of situation awareness," in *International Conference on Neural Information Processing*, vol. 7663 of *Lecture Notes in Computer Science*, pp. 473–483, Springer, Berlin, Germany, 2012.
- [7] M. Castellanos, C. Gupta, S. Wang, U. Dayal, and M. Durazo, "A platform for situational awareness in operational BI," *Decision Support Systems*, vol. 52, no. 4, pp. 869–883, 2012.
- [8] T. Bosse, K. Majdanik, K. Boersma, and K. Ingibergsdóttir, "Studying shared situation awareness by agent-based simulation," in *Proceedings of the 12th IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2013*, pp. 201–208, Atlanta, Ga, USA, November 2013.
- [9] J. C. Lo and S. A. Meijer, "Measuring group situation awareness in a multiactor gaming simulation: A pilot study of railway and passenger traffic operators," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 177–181, 2013.
- [10] J. Li, D. Zhang, and P. Zhang, "Supporting dynamic situation awareness in online group discussion: A visualization approach," in *Proceedings of the 46th Annual Hawaii International Conference on System Sciences, HICSS 2013*, pp. 470–479, Wailea, Maui, Hawaii, USA, January 2013.
- [11] G. Castellano, M. G. C. A. Cimino, A. M. Fanelli, B. Lazzerini, F. Marcelloni, and M. A. Torsello, "A multi-agent system for enabling collaborative situation awareness via position-based stigmergy and neuro-fuzzy learning," *Neurocomputing*, vol. 135, pp. 86–97, 2014.
- [12] G. D'Aniello, V. Loia, and F. Orcioli, "A multi-agent fuzzy consensus model in a Situation Awareness framework," *Applied Soft Computing*, vol. 30, pp. 430–440, 2015.
- [13] T. Bosse and N. Mogles, "Spread of situation awareness in a group: Population-based vs. agent-based modelling," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2014*, pp. 1117–1124, Warsaw, Poland, August 2014.
- [14] D. C. Parker, S. M. Manson, M. A. Janssen, M. J. Hoffmann, and P. Deadman, "Multi-agent systems for the simulation of land-use and land-cover change: a review," *Annals of the Association of American Geographers*, vol. 93, no. 2, pp. 314–337, 2003.
- [15] F. Piette, A. El Fallah, C. Seghrouchni, P. Taillibert, and C. Dinont, "A Multi-Agent Approach for the Deployment of Distributed Applications," in *Smart Environments*, pp. 37–46, Paris, France, 2016.
- [16] J. Dai, Z. Li, and N. Ivanova, "Regional water resource allocation problem in multi-agent game: Taking the water rights transfer as example," *Advances in Intelligent Systems and Computing*, vol. 502, pp. 803–815, 2017.
- [17] M. Othmani-Guibourg, A. E. Fallah-Seghrouchni, J.-L. Farges, and M. Potop-Butucaru, "Multi-agent patrolling in dynamic environments," in *Proceedings of the 2017 IEEE International Conference on Agents, ICA 2017*, pp. 72–77, Beijing, China, July 2017.
- [18] J. Liu, K. M. Kockelman, P. M. Boesch, and F. Ciari, "Tracking a system of shared autonomous vehicles across the Austin, Texas network using agent-based simulation," *Transportation*, vol. 44, no. 6, pp. 1261–1278, 2017.
- [19] H. Aslam, A. Sidorov, N. Bogomazov, F. Bereznyuk, and J. A. Brown, "Relief camp manager: A serious game using the world health organization's relief camp guidelines," in *European Conference on the Applications of Evolutionary Computation*, vol. 10199 of *Lecture Notes in Computer Science*, pp. 407–417, Springer, 2017.
- [20] I.-H. Chen, "An evolutionary game study of an ecological industry chain based on multi-agent simulation: A case study of the Poyang Lake Eco-Economic zone," *Sustainability*, vol. 9, no. 7, article 1165, 2017.
- [21] J. Sušnik, C. Chew, X. Domingo et al., "Multi-stakeholder development of a serious game to explore the water-energy-food-land-climate nexus: The SIM4NEXUS approach," *Water*, vol. 10, no. 2, p. 139, 2018.
- [22] S. W. Jaffry and J. Treur, "Agent-based and population-based modeling of trust dynamics," in *Transactions on Computational Collective Intelligence IX*, vol. 7770 of *Lecture Notes in Computer Science*, pp. 124–151, Springer, Berlin, Germany, 2013.
- [23] T. Bosse, C. Gerritsen, M. Hoogendoorn, S. W. Jaffry, and J. Treur, "Agent-based vs. population-based simulation of displacement of crime: A comparative study," *Web Intelligence and Agent Systems*, vol. 9, no. 2, pp. 147–160, 2011.
- [24] T. Bosse, S. W. Jaffry, G. F. Siddiqui, and J. Treur, "Comparative analysis of agent-based and population-based modelling in epidemics and economics," *Multiagent and Grid Systems*, vol. 8, no. 3, pp. 223–255, 2012.
- [25] R. Aydoğlan, A. Sharpanskykh, and J. Lo, "A Trust-Based Situation Awareness Model," in *European Conference on Multi-Agent Systems*, vol. 8953 of *Lecture Notes in Computer Science*, pp. 19–34, Springer, 2015.
- [26] J. O'Donovan, R. E. Jones, L. R. Marusich, Y. Teng, C. Gonzalez, and T. Höllerer, "A model-based evaluation of trust and situation awareness in the diner's dilemma game," in *Proceedings of the 22th Behavior Representation in Modeling & Simulation (BRIMS) Conference*, pp. 11–14, Ottawa, Canada, 2013.
- [27] K. Sonoda and T. Wada, "Displaying system situation awareness increases driver trust in automated driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 185–193, 2017.

- [28] Z. Nasar and S. W. Jaffry, "Trust-based situation awareness: Agent-based versus population-based modeling a comparative study," in *Proceedings of the 2018 International Conference on Advancements in Computational Sciences (ICACS)*, pp. 1–7, Lahore, Pakistan, February 2018.
- [29] T. Bosse and N. M. Mogles, "Formal analysis of aviation incidents," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, vol. 7345 of *Lecture Notes in Computer Science*, pp. 371–380, Springer, Berlin, Germany, 2012.

Research Article

Improved Optimization for Wastewater Treatment and Reuse System Using Computational Intelligence

Zong Woo Geem ¹, Sung Yong Chung,² and Jin-Hong Kim ²

¹Department of Energy IT, Gachon University, Seongnam 13120, Republic of Korea

²Department of Civil & Environmental Engineering, Chung-Ang University, Seoul 06974, Republic of Korea

Correspondence should be addressed to Jin-Hong Kim; jinhong.kim.cau@gmail.com

Received 1 September 2017; Accepted 5 February 2018; Published 29 April 2018

Academic Editor: Jin H. Yoon

Copyright © 2018 Zong Woo Geem 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.

River water pollution by wastewater can cause significant negative impact on the aquatic sustainability. Hence, accurate modeling of this complicated system and its cost-effective treatment and reuse decision is very important because this optimization process is related to economic expenditure, societal health, and environmental deterioration. In order to optimize this complex system, we may consider three treatment or reuse options such as microscreening filtration, nitrification, and fertilization-oriented irrigation on top of two existing options such as settling and biological oxidation. The objective of this environmental optimization is to minimize the economic expenditure of life cycle costs while satisfying the public health standard in terms of groundwater quality and the environmental standard in terms of river water quality. Particularly, this study improves existing optimization model by pinpointing the critical deficit location of dissolved oxygen sag curve by using analytic differentiation. Also, the proposed formulation considers more practical constraints such as maximal size of irrigation area and minimal amount of filtration treatment process. The results obtained by using an evolutionary algorithm, named a parameter-setting-free harmony search algorithm, show that the proposed model successfully finds optimal solutions while conveniently locating the critical deficit point.

1. Introduction

Polluted water dumped into a river causes considerable negative impact on the sustainability of aquatic life. Thus, accurate modeling of this complex system and optimal decision making is therefore very critical since this relates to environmental deterioration, economic expenditure, and human health.

In order to optimize this complicated problem, we should first consider various life cycle costs of wastewater treatment and reuse alternatives such as microscreening filtration, nitrification, and diverted fertilization-wise irrigation as well as settling and biological oxidation [1]. Under the optimally minimized budget, all the technical and regulatory constraints such as dissolved oxygen level along the river reach, nitrate-nitrogen level in groundwater, nitrogen amount for crop uptake, irrigation area size, and treatment amount in filtration process should be addressed.

To date, various simulation-optimization approaches have been proposed to the wastewater treatment problem

[1–7]. However, there is still more room to improve in terms of optimization formulation and simulation calculation for obtaining better solutions. Particularly, this study improves existing simulation calculations by exactly finding the critical deficit point of dissolved oxygen sag curve using analytic calculus. While previous approaches [1–3] approximate the minimal point by roughly discretizing the continuous river reach into eight points (5, 10, 15, 20, 25, 30, 40, and 50 km away from the effluent spot), this study clearly pinpoints the minimal location by analytically performing the differential calculus. In addition, the proposed formulation considers more practical constraints such as maximal size of irrigated area and minimal amount of filtration treatment.

Thus, this study intends to improve the optimization structure and the calculation process of the complex wastewater treatment and reuse problem and then to find the optimal solution using an improved evolutionary algorithm, which does not require the boring process of algorithm parameter setting.

TABLE 1: Quantified effect of each treatment.

Type of treatment	Effluent quality (in ppm)		
	DO	CBOD	NBOD
Secondary process only (Q_1)	2	25	54
Secondary + filtration (Q_2)	2	13	50
Secondary + nitrification ($Q_3 - Q_4$)	2	13	10
Secondary + nitrification + filtration (Q_4)	2	7	10

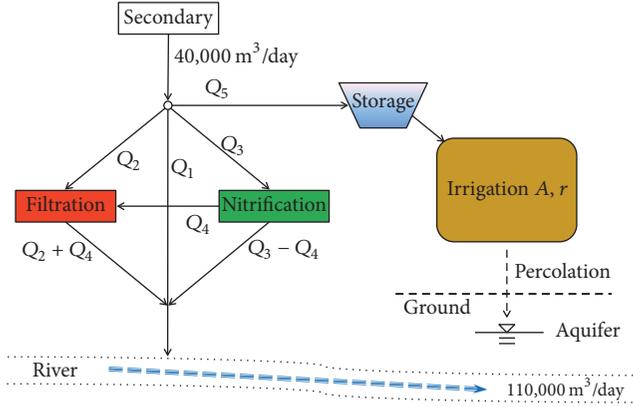


FIGURE 1: Schematic of wastewater treatment system.

2. Problem and Modeling

The original wastewater treatment and reuse problem was proposed by Haith [1]. A municipal area, where 100,000 citizens reside, dumps wastewater ($40,000 \text{ m}^3/\text{day}$) into a river after treating it with settling and biological oxidation. However, these processes are not enough to meet the water quality regulation (5 ppm or 5 mg/liter in terms of dissolved oxygen (DO)). Thus, the local government plans to construct an extra treatment system to improve the effluent quality.

As seen in Figure 1, the extra treatment system can include two treatment options (filtration and nitrification) and one diverted irrigation option for the sake of crop fertilization. In this system, there are five decision variables such as (1) wastewater volume directly dumped into the river (Q_1 kilo- m^3/day), (2) wastewater volume treated by filtration process (Q_2 kilo- m^3/day), (3) wastewater volume treated by nitrification process (Q_3 kilo- m^3/day), (4) wastewater volume treated by nitrification and filtration processes (Q_4 kilo- m^3/day), and (5) wastewater volume diverted for crop fertilization (Q_5 kilo- m^3/day).

For these extra treatment options (doing nothing (Q_1), filtration only (Q_2), and nitrification only ($Q_3 - Q_4$) and both nitrification and filtration (Q_4)), quantified effects can be assessed as described in Table 1.

If wastewater is fully mixed with river flow at the discharge point, initial effluent quality C_0 , B_0 , and N_0 can be calculated using Table 1 and weighted average. If river flow is $110,000 \text{ m}^3/\text{day}$, DO is 8.0 ppm, carbonaceous biochemical oxygen demand (CBOD) is 2.0 ppm, and nitrogenous

biochemical oxygen demand (NBOD) is 5.0 ppm, C_0 , B_0 , and N_0 become as in

$$C_0 = \frac{8(110) + 2(Q_1 + Q_2 + Q_3)}{110 + Q_1 + Q_2 + Q_3},$$

$$B_0 = \frac{2(110) + 25Q_1 + 13Q_2 + 13(Q_3 - Q_4) + 7Q_4}{110 + Q_1 + Q_2 + Q_3}, \quad (1)$$

$$N_0 = \frac{5(110) + 54Q_1 + 50Q_2 + 10(Q_3 - Q_4) + 10Q_4}{110 + Q_1 + Q_2 + Q_3}.$$

For the discharged amount ($Q_1 + Q_2 + Q_3$) into the river, dissolved oxygen level $C(x)$ at x km from the discharge point can be expressed as in (2). Here, u is flow velocity (this study uses 7.9 km/day); k_2 is reaeration rate (this study uses 0.5/day); C_s is saturation DO (this study uses 8.0 ppm); B and N are CBOD and NBOD (ppm) at x ; and k_1 and k_n are rate constants (this study uses 0.35/day and 0.2/day, resp.)

$$u \frac{dC}{dx} = k_2(C_s - C) - k_1B - k_nN. \quad (2)$$

The first term of the right-hand side (RHS) in (2) stands for oxygen increase owing to reaeration and the second and third terms stand for oxygen decrease owing to oxidation by carbonaceous and nitrogenous material.

The differential equation can have an analytic solution by manipulating it using (3) to (14) [3]. From the relationship

$$u \frac{dB}{dx} = -k_1B,$$

$$u \frac{dN}{dx} = -k_nN, \quad (3)$$

(2) becomes

$$\frac{dC}{dx} = \frac{k_2}{u}(C_s - C) - \frac{k_1}{u}B_0e^{-k_1x/u} - \frac{k_n}{u}N_0e^{-k_nx/u}. \quad (4)$$

Equation (4) can be written again in the form of 1st-order nonhomogeneous ordinary differential equation as in

$$\frac{dC}{dx} + \frac{k_2}{u}C = q(x), \quad (5)$$

$$q(x) = \frac{k_2}{u}C_s - \frac{k_1}{u}B_0e^{-k_1x/u} - \frac{k_n}{u}N_0e^{-k_nx/u}.$$

If an integral factor μ is introduced,

$$\mu = e^{\int (k_2/u)dx} = e^{(k_2/u)x}. \quad (6)$$

From (5) and (6), (7) is derived because $d\mu/dx = (k_2/u)\mu$

$$\mu \frac{dC}{dx} + C \frac{d\mu}{dx} = \mu q(x). \quad (7)$$

Equation (7) can be further rewritten as (8)

$$\frac{d}{dx} (\mu C) = \mu q(x). \quad (8)$$

Equation (8) can be integrated as (9)

$$\mu C = \int \mu q(x) dx + \widetilde{C}. \quad (9)$$

If Equation (9) is divided by μ , it becomes

$$\begin{aligned} C &= \frac{1}{\mu} \int \mu q(x) dx + \frac{1}{\mu} \widetilde{C} \\ &= e^{-(k_2/u)x} \int e^{(k_2/u)x} q(x) dx + \widetilde{C} e^{-(k_2/u)x}. \end{aligned} \quad (10)$$

Here, $\int e^{(k_2/u)x} q(x) dx$ becomes

$$\begin{aligned} \int \left(\frac{k_2}{u} C_s e^{(k_2/u)x} - \frac{k_1}{u} B_0 e^{(-k_1/u)x} e^{(k_2/u)x} \right. \\ \left. - \frac{k_n}{u} N_0 e^{(-k_n/u)x} e^{(k_2/u)x} \right) dx = C_s e^{(k_2/u)x} \\ - \frac{k_1}{k_2 - k_1} B_0 e^{((k_2 - k_1)/u)x} - \frac{k_n}{k_2 - k_n} N_0 e^{((k_2 - k_n)/u)x} \\ + \widetilde{C}. \end{aligned} \quad (11)$$

Thus, (10) becomes

$$\begin{aligned} C &= C_s - \frac{k_1}{k_2 - k_1} B_0 e^{(-k_1/u)x} - \frac{k_n}{k_2 - k_n} N_0 e^{(-k_n/u)x} \\ &+ \widetilde{C} e^{(-k_2/u)x} + \widetilde{C} e^{(-k_2/u)x}. \end{aligned} \quad (12)$$

If $x = 0$ and $D = \widetilde{C} + \widetilde{C}$, (12) becomes

$$C_0 = C_s - \frac{k_1 B_0}{k_2 - k_1} - \frac{k_n N_0}{k_2 - k_n} + D. \quad (13)$$

Finally, an analytic solution is obtained as in (14), which is known as the DO sag equation by Streeter–Phelps [8]

$$\begin{aligned} C(x) &= C_s - \frac{k_1 B_0}{k_2 - k_1} e^{(-k_1/u)x} - \frac{k_n N_0}{k_2 - k_n} e^{(-k_n/u)x} \\ &+ \left(C_0 - C_s + \frac{k_1 B_0}{k_2 - k_1} + \frac{k_n N_0}{k_2 - k_n} \right) e^{(-k_2/u)x} \\ &= C_s \left(1 - e^{(-k_2/u)x} \right) + C_0 e^{(-k_2/u)x} \\ &- \frac{k_1 B_0}{k_2 - k_1} \left(e^{(-k_1/u)x} - e^{(-k_2/u)x} \right) \\ &- \frac{k_n N_0}{k_2 - k_n} \left(e^{(-k_n/u)x} - e^{(-k_2/u)x} \right). \end{aligned} \quad (14)$$

Another option, instead of the filtration and/or nitrification treatments, is diverted irrigation for crop fertilization.

The diverted volume Q_5 has nitrogen. If the nitrogen concentration of Q_5 is n (20 ppm in this study), the contained nitrogen amount becomes $0.1rTn$ (kg/ha). Here, 0.1 is unit conversion coefficient; r is another decision variable, denoting irrigation rate (cm/week); and T is the irrigation period (13 weeks in this study). If the nitrogen consuming amount of crop is NC (170 kg/ha in this study), the nitrogen amount ($0.1rTn$) in Q_5 should be equal to or greater than the crop's nitrogen demand (NC) as follows:

$$0.1rTn \geq \text{NC}. \quad (15)$$

The surplus nitrogen amount $0.1rTn - \text{NC}$ is percolated into groundwater. And, total liquid amount percolated into groundwater becomes $rT + P - \text{ET}$. Here, P is precipitation amount (cm) during the irrigation period; and ET is evapotranspiration amount (cm) during the irrigation period. Thus, the nitrate-nitrogen concentration c_n (ppm) in the percolation amount can be expressed as in (16). Here, 10 is the unit conversion coefficient; and c_n should be equal to or less than 10 ppm according to local government regulation

$$c_n = \frac{\text{surplus nitrogen}}{\text{percolation}} = \frac{10(0.1rTn - \text{NC})}{rT + P - \text{ET}}. \quad (16)$$

3. Improved Optimization Formulation

The objective function of the wastewater treatment problem described in the previous section is the total life cycle cost to be minimized as in (17) [1, 2]. The cost function consists of three subcosts including filtration process cost C_{ft} , nitrification process cost C_{nt} , and irrigation process cost C_{ir} .

$$\begin{aligned} \text{Minimize } z &= f(Q_1, Q_2, Q_3, Q_4, Q_5, r) \\ &= C_{\text{ft}} + C_{\text{nt}} + C_{\text{ir}}. \end{aligned} \quad (17)$$

C_{ft} ($\$10^3/\text{year}$) is composed of capital cost (first term in RHS of (18)) and operation and maintenance (O&M) cost (second term in RHS of (18))

$$C_{\text{ft}} = 3(Q_2 + Q_4)^{0.93} + 6.7(Q_2 + Q_4)^{0.55}. \quad (18)$$

C_{nt} ($\$10^3/\text{year}$) is composed of capital cost (first term in RHS of (19)) and O&M cost (second term in RHS of (19))

$$C_{\text{nt}} = 13.8Q_3^{0.68} + 10.6Q_3^{0.42}. \quad (19)$$

C_{ir} ($\$10^3/\text{year}$) is composed of capital cost of transmission pipeline (first term in RHS of (20)), capital cost of storage system (second term in RHS of (20)), O&M cost of storage system (third term in RHS of (20)), capital cost of irrigation system (4th term in RHS of (20)), O&M cost of irrigation system (5th term in RHS of (20)), land rent cost (6th term in RHS of (20)), and crop sales benefit (7th term in RHS of (20)),

(20)). Here, A is irrigated area (ha), which can be calculated as $70Q_5/r$

$$\begin{aligned} C_{ir} = & 21.9Q_5^{0.28} + 1.2Q_5^{0.78} + 0.2Q_5^{0.54} \\ & + \left(13.1 + \frac{48}{r}\right) Q_5^{(0.74+0.32/r)} \\ & + \left(5.1 + \frac{19}{r}\right) Q_5^{(0.79+0.28/r)} + 0.19A - 0.87A. \end{aligned} \quad (20)$$

Technical and regulatory constraints for this problem are

$$Q_1 + Q_2 + Q_3 + Q_5 = 40, \quad (21)$$

$$Q_3 \geq Q_4, \quad (22)$$

$$0 \leq Q_i \leq 40, \quad i = 1, \dots, 5, \quad (23)$$

$$\begin{aligned} & 8 \left(1 - e^{-0.063x}\right) + C_0 e^{-0.063x} \\ & - 2.33B_0 \left(e^{-0.044x} - e^{-0.063x}\right) \\ & - 0.67N_0 \left(e^{-0.025x} - e^{-0.063x}\right) \geq 5, \end{aligned} \quad (24)$$

$$6.54 \leq r \leq 13.07. \quad (25)$$

Equation (21) constrains the total wastewater amount to be equal to 40 kilo m^3 /day; (22) constrains the amount of nitrification to be equal to or greater than that of both nitrification and filtration; (23) constrains any subwastewater amount to stay between lower and upper limits (0 to 40 kilo m^3 /day); (24) constrains DO level at any point along the river reach to be greater than or equal to 5 ppm; and (25) constrains total nitrogen amount to be greater than or equal to crop nitrogen requirement as specified in (15) and also constrains nitrate-nitrogen concentration c_n to be less than or equal to 10 ppm as specified in (16).

This wastewater treatment optimization problem was originally proposed by Haith [1] and more optimally solved using harmony search [2]. However, total discharged amount is assigned only to irrigation ($Q_5 = 32.19$ kilo m^3 /day) or no further treatment ($Q_1 = 7.81$ kilo m^3 /day) while nothing is assigned to filtration ($Q_2 = 0$) and nitrification ($Q_3 = 0$) because this way is more cost-efficient [2]. The best solution is $f(Q_1 = 7.81, Q_2 = 0, Q_3 = 0, Q_4 = 0, Q_5 = 32.19, r = 7.40) = 303.0$ which satisfies all the constraints in (21) to (25).

In order to improve this biased result, a slight modification was made to the cost coefficients [3]. When the cost coefficients for filtration and nitrification were reduced, some portion was assigned to Q_2 and Q_3 [3]. However, this case is also not very realistic because most of the portion was assigned to Q_2 and Q_3 while only a tiny amount was assigned to Q_5 and Q_1 .

Here, if we analyze the best solution in [2], we may improve the optimization formulation. For example, without the constraint of irrigated area, the area size can be expanded up to 428 ha. But, more realistically we can consider the maximum irrigation size (for example, 100 ha in this study), as follows:

$$A \leq 100. \quad (26)$$

Also, the filtration process may require minimum treatment amount (e.g., 10 kilo m^3 /day in this study) because a tiny amount is not realistic for operating the treatment plant cost-effectively, as follows:

$$Q_2 + Q_4 \geq 10. \quad (27)$$

For the DO level constraint specified in (24), we have to ideally check the entire river reach. However, we were not able to check all the continuous points because the number of tasks is astronomical. Thus, previous research [1–3] approximately and arbitrarily checked only 8 points (5, 10, 15, 20, 25, 30, 40, and 50 km from the origin). This approach is not accurate at all and sometimes violates minimal DO level because actual maximum deficit point exists between two points. Also, there is a chance that the deficit point exists out of the range (5 to 50 km) if the model is applied to other problems.

In order to overcome this critical drawback, this study proposes an analytical solution which can deterministically find the minimal DO location instead of checking only eight points. Basically, we can obtain the minimal DO location by differentiating the DO function specified in (14). However, the task is too complicated to easily obtain it. Nonetheless, this very problem has a problem-specific way to find the location deterministically.

Equation (14) can be differentiated as follows:

$$\begin{aligned} \frac{d}{dx} C(x) = & -\alpha e^{-(k_2/u)x} + \beta e^{-(k_1/u)x} + \gamma e^{-(k_n/u)x}, \\ \alpha = & \frac{k_2}{u} \left(C_0 - C_s + \frac{k_1 B_0}{k_2 - k_1} + \frac{k_n N_0}{k_2 - k_n} \right), \\ \beta = & \frac{k_1}{u} \frac{k_1 B_0}{k_2 - k_1}, \\ \gamma = & \frac{k_n}{u} \frac{k_n N_0}{k_2 - k_n}. \end{aligned} \quad (28)$$

To solve (28), we introduce $e^{(k_2/u)x}$ which is greater than 0.

$$e^{(k_2/u)x} \left(-\alpha e^{-(k_2/u)x} + \beta e^{-(k_1/u)x} + \gamma e^{-(k_n/u)x} \right) = 0. \quad (29)$$

It further becomes

$$-\alpha + \beta e^{((k_2-k_1)/u)x} + \gamma e^{((k_2-k_n)/u)x} = 0. \quad (30)$$

For this specific problem, $k_2 = 0.5$ /day, $k_1 = 0.35$ /day, and $k_n = 0.2$ /day. Thus, the following equation is satisfied:

$$\frac{k_2 - k_n}{u} x = 2 \frac{k_2 - k_1}{u} x. \quad (31)$$

If $z = e^{((k_2-k_1)/u)x}$, (30) becomes

$$-\alpha + \beta z + \gamma z^2 = 0. \quad (32)$$

Hence, using quadratic formula, z becomes

$$z = \frac{-\beta \pm \sqrt{\beta^2 + 4\alpha\gamma}}{2\gamma}. \quad (33)$$

Since $z = e^{((k_2 - k_1)/u)x} > 0$,

$$z = e^{((k_2 - k_1)/u)x} = \frac{-\beta + \sqrt{\beta^2 + 4\alpha\gamma}}{2\gamma}. \quad (34)$$

Thus,

$$\frac{k_2 - k_1}{u} x = \ln \left(\frac{-\beta + \sqrt{\beta^2 + 4\alpha\gamma}}{2\gamma} \right). \quad (35)$$

Finally,

$$x = \frac{u}{k_2 - k_1} \ln \left(\frac{-\beta + \sqrt{\beta^2 + 4\alpha\gamma}}{2\gamma} \right). \quad (36)$$

4. Harmony Search Algorithm

This study utilizes a parameter-setting-free harmony search (PSF-HS) algorithm for obtaining the optimal solution. Different from the original harmony search (HS) algorithm, the

$$x_i^{\text{New}} \leftarrow \begin{cases} x_i \in [x_i^{\text{Lower}}, x_i^{\text{Upper}}] \\ x_i \in \text{HM} = \{x_i^1, x_i^2, \dots, x_i^{\text{HMS}}\} \\ x_i + \Delta, x_i \in \text{HM} \end{cases}$$

where Δ is pitch adjusting volume that is obtained as $(x_i^{\text{Upper}} - x_i^{\text{Lower}})/1000 \times u(-1, 1)$ in this study; $u(-1, 1)$ is uniform random number generated between -1 and 1 ; HMCR stands for harmony memory considering rate (value range: $0 \leq \text{HMCR} \leq 1$); and PAR stands for pitch adjusting rate (value range: $0 \leq \text{PAR} \leq 1$).

If the new harmony \mathbf{x}^{New} satisfies every constraint and is better than the worst harmony $\mathbf{x}^{\text{Worst}}$ contained in HM, the former is included in HM and the latter is excluded from HM as follows:

$$\mathbf{x}^{\text{New}} \in \text{HM} \wedge \mathbf{x}^{\text{Worst}} \notin \text{HM}. \quad (39)$$

The processes specified in (38) and (39) are repeated until a termination criterion, such as maximum number of iterations (improvisations), is satisfied.

The PSF-HS algorithm, when compared with the basic HS algorithm, has one extra matrix, called operation type matrix (OTM). Equation (40) shows one example of OTM

$$x_i^{\text{New}} \leftarrow \begin{cases} x_i \in [x_i^{\text{Lower}}, x_i^{\text{Upper}}] \\ x_i \in \text{HM} = \{x_i^1, x_i^2, \dots, x_i^{\text{HMS}}\} \\ x_i + \Delta, x_i \in \text{HM} \end{cases}$$

PSF-HS algorithm does not require the tedious process of algorithm parameter value setting [9] and has so far been applied to various bench-mark problems [10, 11] and engineering optimization problems including structural design [12] and groundwater pollution source identification [13]. More generally, the HS algorithm has so far been theoretically developed and practically applied to various computational intelligence and complex problems [14–16].

The basic structure of HS algorithm contains a solution pool named harmony memory (HM), which has randomly generated solutions as many as harmony memory size (HMS; 30 in this study) [2]:

$$\text{HM} = \begin{bmatrix} Q_1^1 & Q_2^1 & Q_3^1 & Q_4^1 & Q_5^1 & r^1 & f(\mathbf{x}^1) \\ Q_1^2 & Q_2^2 & Q_3^2 & Q_4^2 & Q_5^2 & r^2 & f(\mathbf{x}^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_1^{\text{HMS}} & Q_2^{\text{HMS}} & Q_3^{\text{HMS}} & Q_4^{\text{HMS}} & Q_5^{\text{HMS}} & r^{\text{HMS}} & f(\mathbf{x}^{\text{HMS}}) \end{bmatrix}. \quad (37)$$

And, at every improvisation (iteration), a new harmony (solution vector) \mathbf{x}^{New} is generated based upon HM as follows:

$$\begin{aligned} \text{w.p.} & \quad (1 - \text{HMCR}) \\ \text{w.p.} & \quad \text{HMCR} \cdot (1 - \text{PAR}) \quad i = 1, \dots, 6 \\ \text{w.p.} & \quad \text{HMCR} \cdot \text{PAR}, \end{aligned} \quad (38)$$

OTM

$$= \begin{bmatrix} o_1^1 = \text{Random} & o_2^1 = \text{Pitch} & \dots & o_6^1 = \text{Memory} \\ o_1^2 = \text{Memory} & o_2^2 = \text{Memory} & \dots & o_6^2 = \text{Pitch} \\ \vdots & \dots & \dots & \dots \\ o_1^{\text{HMS}} = \text{Memory} & o_2^{\text{HMS}} = \text{Random} & \dots & o_6^{\text{HMS}} = \text{Memory} \end{bmatrix}. \quad (40)$$

OTM memorizes the operation history of every harmony stored in HM. At the early stage, HMCR and PAR have a constant value of 0.5, and OTM is accumulating the operation information. Then, HMCR and PAR are iteratively updated based on OTM, instead of using fixed parameter values

$$\begin{aligned} \text{HMCR}_i &= \frac{\text{ct}(o_i^j = \text{Memory} \vee o_i^j = \text{Pitch}, \forall j)}{\text{HMS}}, \\ \text{PAR}_i &= \frac{\text{ct}(o_i^j = \text{Pitch}, \forall j)}{\text{ct}(o_i^j = \text{Memory} \vee o_i^j = \text{Pitch}, \forall j)}, \end{aligned} \quad (41)$$

where $\text{ct}(\cdot)$ stands for the function which returns the number of designated operations.

Since every decision variable has different HMCR and PAR, (38) can be slightly changed as follows:

$$\begin{aligned} \text{w.p.} & \quad (1 - \text{HMCR}_i) \\ \text{w.p.} & \quad \text{HMCR}_i \cdot (1 - \text{PAR}_i) \quad i = 1, \dots, 6 \\ \text{w.p.} & \quad \text{HMCR}_i \cdot \text{PAR}_i, \end{aligned} \quad (42)$$

TABLE 2: Optimization results for wastewater treatment system.

Run	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Cost (k\$/yr)	373.1	375.5	377.6	375.4	383.1	378.2	377.7	371.5	377.2	379.1
Q_1 (k ton/d)	0.12	0.01	0.13	0.15	0.48	0.16	0.10	0.05	0.49	0.37
Q_2 (k ton/d)	0.12	0.30	0.61	0.26	0.84	0.70	0.65	0.05	0.11	0.50
Q_3 (k ton/d)	31.95	32.34	30.55	31.64	28.67	30.08	30.48	32.37	30.67	30.01
Q_4 (k ton/d)	9.96	11.33	9.75	10.38	9.68	9.40	9.67	9.95	9.97	9.75
Q_5 (k ton/d)	7.81	7.35	8.71	7.96	10.01	9.07	8.77	7.53	8.73	9.13
r (cm/wk)	13.07	13.07	13.07	13.07	13.07	13.07	13.07	13.07	13.07	13.07
$Q_2 + Q_4$ (≥ 10)	10.08	11.63	10.36	10.64	10.52	10.10	10.32	10.00	10.08	10.24
$Q_3 - Q_4$ (≥ 0)	21.99	21.01	20.80	21.26	18.99	20.68	20.81	22.43	20.70	20.26
A (≤ 100 ha)	41.9	39.4	46.7	42.6	53.6	48.6	46.9	40.3	46.8	48.9
$0.1rTn$ (≥ 170 kg/ha)	339.8	339.8	339.8	339.8	339.8	339.8	339.8	339.8	339.8	339.8
c_n (≤ 10 ppm)	9.99	9.99	9.99	9.99	9.99	9.99	9.99	9.99	9.99	9.99
x (km)	16.79	16.75	16.95	16.83	17.16	17.00	16.95	16.75	16.94	17.01
$C(x)$ (≥ 5 ppm)	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00

5. Computation Results

The improved optimization model was applied to the wastewater treatment and reuse system and obtained the results using the PSF-HS approach (5,000 iterations). Table 2 shows the detailed results out of 10 runs. While a traditional mathematical optimization algorithm such as generalized reduced gradient method gets stuck in local optima or even diverges [2], the PSF-HS was able to robustly find solution vectors.

As observed in the table, optimal costs were obtained, ranging from 371.5×10^3 /year to 383.1×10^3 /year with the average of 376.8×10^3 /year. Also, all the constraints were satisfied: every filtration inflow ($Q_2 + Q_4$) was greater than minimally required amount (10 kilo-ton/day); every pure nitrification amount ($Q_3 - Q_4$) was greater than 0, which means $Q_3 \geq Q_4$; every irrigation area is less than 100 ha; every nitrogen amount for crop fertilization is greater than the required amount (170 kg/ha); every water quality in groundwater is less than the regulated level (10 ppm); and every DO level at the pinpointed lowest location is greater than the minimally required level (5 ppm).

Figure 2 shows DO sag curve for the best cost ($\$371.5 \times 10^3$ /year) from the solution vector ($Q_1 = 0.05$; $Q_2 = 0.05$; $Q_3 = 32.37$; $Q_4 = 9.95$; $Q_5 = 7.53$; $r = 13.07$). The

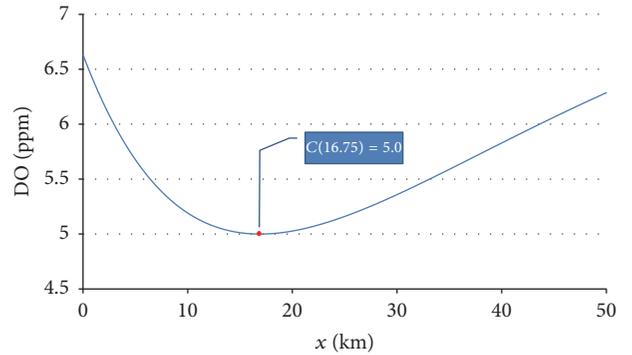


FIGURE 2: Dissolved oxygen sag curve.

analytic solution specified in (36) could find the critical deficit location (16.75 km) of DO more easily and efficiently for this problem.

Figures 3 and 4 show the histories of HMCR and PAR values in obtaining the best solution ($\$371.5 \times 10^3$ /year). As observed in Figure 3, all the HMCR values have converged into high values (≥ 0.95) after 600 iterations. This means that PSF-HS conservatively depends on the values stored in HM.

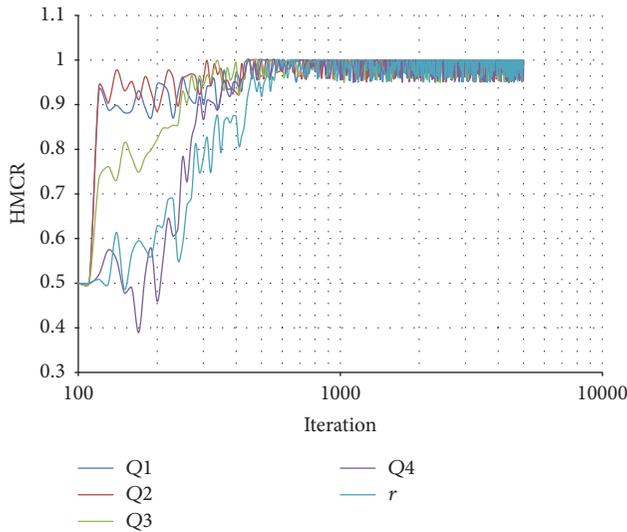


FIGURE 3: History of HMCR for best solution.

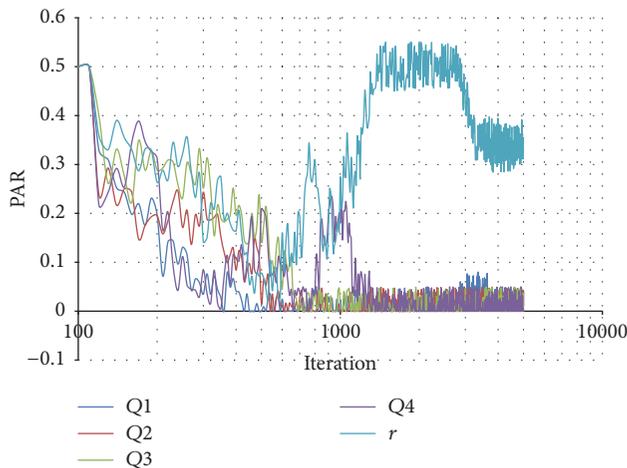


FIGURE 4: History of PAR for best solution.

Meanwhile, as observed in Figure 4, all PAR have converged into low values (≤ 0.1) after 1,000 iterations, except the PAR for irrigation rate r , which rose up to 0.5 and then slightly went down to 0.35. This means that PSF-HS mostly does not need any further pitch adjustment for each treated or diverted volume; however, the irrigation rate needs to be frequently pitch-adjusted because it should satisfy the nitrate-nitrogen concentration level ($c_n \leq 10$ ppm) in groundwater.

6. Conclusions

This study proposed an improved optimization model for the wastewater treatment and reuse system that contains three options such as filtration, nitrification, and diverted irrigation and found optimal solutions using the PSF-HS technique. The improved optimization model could obtain the minimal life cycle cost design while satisfying DO level at the critical deficit location, minimal amount for filtration process, and

maximal size of irrigation area as well as proper groundwater quality and nitrogen amount for crop farming.

The major improvements of this study are to consider more realistic design constraints such as maximal irrigation size and minimal filtration amount and to pinpoint the critical deficit location of dissolved oxygen by using differential calculus instead of checking roughly dispersed points. While previous approaches checking eight discrete points had difficulty in finding the exact location of the critical deficit point, this study could exactly spot the point using analytic solution of the DO concentration function.

However, the weakness of this study is the fact that the analytic solution can be obtained only in special cases where the difference amount between reaeration rate (k_2) and NBOD rate constant (k_n) is twice as large as that between reaeration rate (k_2) and CBOD rate constant (k_1), observed in this bench-mark problem. Thus, for future research, we would like to develop the analytic solution which can be applied to more general cases.

With respect to algorithm side, PSF-HS has advantages over original HS because it does not demand a tedious parameter setting process for HMCR and PAR and individually considers HMCR and PAR for each variable [9]. Meanwhile, PSF-HS also has disadvantages because it has an additional matrix and requires additional setting (starting HMCR and PAR values and duration of rehearsal). Also, the solution quality of PSF-HS is not always better than that of original HS whose parameter values are manually and properly chosen [9]. Thus, the future research direction of PSF-HS can be how to efficiently perform the additional setting process and how to enhance the solution quality.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by a grant from Advanced Water Management Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government (12-TI-C02) and by the Chung-Ang University Research Scholarship Grants in 2015.

References

- [1] D. A. Haith, *Environmental Systems Optimization*, Wiley, New York, NY, USA, 1982.
- [2] Z. W. Geem and J.-H. Kim, "Wastewater treatment optimization for fish migration using harmony search," *Mathematical Problems in Engineering*, vol. 2014, Article ID 313157, 5 pages, 2014.
- [3] Z. W. Geem and J.-H. Kim, "Sustainable optimization for wastewater treatment system using PSF-HS," *Sustainability*, vol. 8, no. 4, Article ID 321, 2016.
- [4] D. H. Burn and B. J. Lence, "Comparison of optimization formulations for waste-load allocations," *Journal of Environmental Engineering*, vol. 118, no. 4, pp. 597–613, 1992.
- [5] L. Somlyódy, M. Kularathna, and I. Masliev, "Development of least-cost water quality control policies for the Nitra River Basin

- in Slovakia,” *Water Science and Technology*, vol. 30, no. 5, pp. 69–78, 1994.
- [6] A. P. Singh, S. K. Ghosh, and P. Sharma, “Water quality management of a stretch of river Yamuna: an interactive fuzzy multi-objective approach,” *Water Resources Management*, vol. 21, no. 2, pp. 515–532, 2007.
- [7] Z. Zhang, A. Kusiak, Y. Zeng, and X. Wei, “Modeling and optimization of a wastewater pumping system with data-mining methods,” *Applied Energy*, vol. 164, pp. 303–311, 2016.
- [8] https://en.wikipedia.org/wiki/Streeter-Phelps_equation.
- [9] Z. W. Geem and K.-B. Sim, “Parameter-setting-free harmony search algorithm,” *Applied Mathematics and Computation*, vol. 217, no. 8, pp. 3881–3889, 2010.
- [10] C.-M. Wang and Y.-F. Huang, “Self-adaptive harmony search algorithm for optimization,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 2826–2837, 2010.
- [11] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, “A self-adaptive global best harmony search algorithm for continuous optimization problems,” *Applied Mathematics and Computation*, vol. 216, no. 3, pp. 830–848, 2010.
- [12] O. Hasançebi, F. Erdal, and M. P. Saka, “Adaptive harmony search method for structural optimization,” *Journal of Structural Engineering*, vol. 136, no. 4, pp. 419–431, 2010.
- [13] S. Jiang, Y. Zhang, P. Wang, and M. Zheng, “An almost-parameter-free harmony search algorithm for groundwater pollution source identification,” *Water Science and Technology*, vol. 68, no. 11, pp. 2359–2366, 2013.
- [14] O. M. Alia and R. Mandava, “The variants of the harmony search algorithm: an overview,” *Artificial Intelligence Review*, vol. 36, no. 1, pp. 49–68, 2011.
- [15] D. Manjarres, I. Landa-Torres, S. Gil-Lopez et al., “A survey on applications of the harmony search algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1818–1831, 2013.
- [16] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, “Harmony search method: theory and applications,” *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 258491, 10 pages, 2015.

Research Article

The Intuitionistic Fuzzy Linguistic Cosine Similarity Measure and Its Application in Pattern Recognition

Donghai Liu ^{1,2}, Xiaohong Chen,^{1,3} and Dan Peng²

¹School of Business, Central South University, Changsha, Hunan 410075, China

²Department of Mathematics, Hunan University of Science and Technology, Xiangtan, Hunan 411201, China

³Hunan University of Commerce, Changsha, Hunan 410205, China

Correspondence should be addressed to Donghai Liu; donghailiu@126.com

Received 7 December 2017; Revised 6 January 2018; Accepted 21 January 2018; Published 22 April 2018

Academic Editor: László T. Kóczy

Copyright © 2018 Donghai 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.

We propose the cosine similarity measures for intuitionistic fuzzy linguistic sets (IFLSs) and interval-valued intuitionistic fuzzy linguistic sets (IVIFLSs), which are expressed by the linguistic scale function based on the cosine function. Then, the weighted cosine similarity measure and the ordered weighted cosine similarity measure for IFLSs and IVIFLSs are introduced by taking into account the importance of each element, and the properties of the cosine similarity measures are also given. The main advantage of the proposed cosine similarity measures is that the decision-makers can flexibly select the linguistic scale function depending on the actual semantic situation. Finally, we present the application of the cosine similarity measures for intuitionistic fuzzy linguistic term sets and interval-valued intuitionistic fuzzy linguistic term sets to pattern recognition and medical diagnosis, and the existing cosine similarity measures are compared with the proposed cosine similarity measures by the illustrative example.

1. Introduction

The fuzzy set was proposed by Zadeh [1] and has achieved a great success in various fields, which is considered to be an effective tool to solve the decision-making problems, pattern recognition, and fuzzy inference [2–4]. Since the fuzzy set was put forward, it was extended in different aspects. One of the generalizations of fuzzy set is intuitionistic fuzzy set (IFS), which was introduced by Atanassov [5]. A typical feature of IFS is that the membership relations are represented by the membership degree and nonmembership degree, respectively. However, due to the fuzziness and uncertainty in the multiple criteria decision-making problems, it is difficult to use the exact values to present qualitative evaluation. At this time, people often provide their opinions in linguistic term sets. In some practical decision-making problems, the decision-maker regards the linguistic information as the values of linguistic variables; that is to say, the values of the variables are not represented by numerical values but are represented by linguistic values, such as “good,” “better,” “fair,” “slightly worse,” and “poor.” Up to now, many people have

studied the problem of linguistic multiple criteria decision-making, Herrera and Verdegay [6] proposed the linguistic assessments in group decision-making (GDM) problem in 1993, then Herrera et al. [7] proposed a consensus model for group decision-making based on linguistic evaluations information, and Herrera et al. [8] considered several group decision-making processes using linguistic ordered weighted averaging (LOWA) operator. Later, Xu [9] proposed a group decision-making method based on the uncertain linguistic ordered weighted geometric (LOWG) operators and the induced uncertain LOWG operators. Furthermore, Xu [10] presented the linguistic hybrid aggregation (LHA) operator and applied it to group decision-making.

However, in some practical decision-making problems, the decision-makers may have some indeterminacy in their linguistic evaluation; they cannot express their preferences by using only membership degree of a linguistic term. Then Wang et al. [11] proposed the intuitionistic fuzzy linguistic aggregation operators and applied them to multicriteria group decision-making problems. For example, $\langle s_3, (0.2, 0.6) \rangle$ is an intuitionistic fuzzy linguistic number

(IFLN), 0.2 is the membership degree of the linguistic term s_3 , and 0.6 is the nonmembership degree of the linguistic term s_3 . The intuitionistic fuzzy linguistic sets (IFLSs) have made great progress in describing linguistic information and to some extent it can be regarded as an innovative construct. The research on this field has been growing rapidly [12–15].

On the other hand, similarity measure is an important topic in the fuzzy set theory, and it is widely used in some fields [16–19], such as pattern recognition, medical diagnosis, and citation analysis. One of the important similarity measures is the cosine similarity measure, which is defined in vector space. Ye [20] proposed the cosine similarity measure and the weighted cosine similarity measure between IFSSs. Zhou et al. [21] presented the intuitionistic fuzzy ordered weighted cosine similarity measure and applied it to the group decision-making problem about the choice of investment plan. Liu et al. [22] presented the interval-valued intuitionistic fuzzy ordered weighted cosine similarity (IVI-FOWCS) measure and applied it to the investment decision-making. As far as we know, the study of cosine similarity measures of intuitionistic fuzzy set has not been discussed. In the following, we will propose the cosine similarity measures of IFLSs and IVIFLSs. The main characteristics of the cosine similarity measures that we can calculate are based on linguistic term set by the linguistic scale function. Linguistic scale function between IFLSs and IVIFLSs was introduced by Wang et al. [23], which was used to calculate the Hausdorff distance between hesitant fuzzy linguistic numbers (HFLNs); it can assign different semantic values to the linguistic terms under different circumstances and improve the flexibility of the proposed cosine similarity measures.

The rest of the paper is organized as follows. In Section 2, some basic concepts of LTSs, IFSSs, IFLSs, and linguistic scale functions are briefly reviewed. In Section 3, we first introduce the cosine similarity measures between IFLSs and then discussed some related properties. Furthermore, the weighted cosine similarity measure between IFLSs, the ordered weighted cosine similarity measure between IFLSs, and the ordered weighted cosine similarity measure between IVIFLSs are analyzed. In Section 4, we give the application of the proposed cosine similarity measures between IFLSs and IVIFLSs on pattern recognition and medical diagnosis and then make comparison analysis with the existing cosine similarity measures. The conclusions are given in Section 5.

2. Preliminaries

In this section, we will review and discuss some related basic concepts, including linguistic term sets (LTSs), intuitionistic fuzzy sets (IFSSs), intuitionistic fuzzy linguistic sets (IFLSs), linguistic scale functions, the ordered weighted averaging (OWA) operator, and the cosine similarity measure between fuzzy sets.

2.1. Linguistic Term Set. In some practical problems, the information expressed by the numerical values may bring inconvenience. At this time it is suitable to use linguistic term set to express information.

Definition 1 (Herrera and Verdegay [6]). Suppose that $S = \{s_\alpha \mid \alpha = 0, 1, \dots, \tau\}$ is a finite and totally ordered discrete term set, where s_α represents a possible value for a linguistic variable; it satisfies the following characteristics:

- (1) $s_\alpha + s_\beta = s_{\alpha+\beta}$
- (2) $\lambda s_\alpha = s_{\lambda\alpha}$
- (3) $s_\alpha > s_\beta$ if $\alpha > \beta$.

For example, a set of seven terms S could be given as follows:

$$\begin{aligned} S &= \{s_0 = \text{very low}, s_1 = \text{low}, s_2 = \text{slightly low}, s_3 \\ &= \text{normal}, s_4 = \text{slightly high}, s_5 = \text{high}, s_6 \\ &= \text{very high}\}. \end{aligned} \quad (1)$$

The discrete linguistic term S cannot usually adapt to the aggregated results. In order to represent these results accurately, Xu [10] extended the discrete term set S to the continuous term set $\bar{S} = \{s_\alpha \mid \alpha \in [0, t]\}$, where t ($t > \tau$) is a sufficiently large positive integer.

2.2. Intuitionistic Fuzzy Set

Definition 2 (Atanassov [5]). Given a fixed set $X = \{x_1, x_2, \dots, x_n\}$, then an intuitionistic fuzzy set ξ in X is defined as

$$\xi = \{(x, \mu_\xi(x), \nu_\xi(x)) \mid x \in X\}, \quad (2)$$

where $\mu_\xi(x)$ ($0 \leq \mu_\xi(x) \leq 1$) and $\nu_\xi(x)$ ($0 \leq \nu_\xi(x) \leq 1$) represent the membership degree and nonmembership degree of x to ξ , respectively, and they satisfy the condition: $0 \leq \mu_\xi(x) + \nu_\xi(x) \leq 1$.

For all $x \in X$, if $\pi_\xi(x) = 1 - \mu_\xi(x) - \nu_\xi(x)$, then $\pi_\xi(x)$ is called the hesitancy degree of x to ξ .

On the basis of intuitionistic fuzzy set and linguistic term set, Wang et al. [11] presented the following intuitionistic fuzzy linguistic term set.

2.3. Intuitionistic Fuzzy Linguistic Term Set

Definition 3 (Wang et al. [11]). Let $X = \{x_1, x_2, \dots, x_n\}$ be a fixed set and $s_{\gamma(x)} \in \bar{S}$ an intuitionistic fuzzy linguistic term set γ in X is defined as

$$\gamma = \{(x, s_{\gamma(x)}, \mu_\gamma(x), \nu_\gamma(x)) \mid x \in X\}, \quad (3)$$

where $\mu_\gamma(x)$ and $\nu_\gamma(x)$ are the membership function and nonmembership function of the element x to $s_{\gamma(x)}$, respectively, and $0 \leq \mu_\gamma(x) + \nu_\gamma(x) \leq 1$ ($x \in X$).

For all $x \in X$, let $\pi_\gamma(x) = 1 - \mu_\gamma(x) - \nu_\gamma(x)$ be the hesitancy function, which means the degree of hesitancy of x to $s_{\gamma(x)}$.

2.4. Linguistic Scale Functions. One of the advantages of linguistic term set is that it can express uncertain information flexibly in practical problems, but if we use the subscript

of linguistic terms directly in the process of operations, it may lose this advantage. The most important thing is to find effective tools to transform linguistic terms to numerical values. As we all know, linguistic scale function (Wang et al. [23]) is a mapping from linguistic term set s_i to the real value η_i . The linguistic scale function can assign different semantic values to the linguistic terms under different circumstances. In practice, the linguistic scale functions are very popular because they are very flexible and they can give more deterministic results based on different semantics.

Definition 4 (Wang et al. [23]). Let $S = \{s_i \mid i = 0, 1, \dots, 2\tau\}$ be a linguistic term. If η_i is a numeric value between 0 and 1, then the linguistic scale function f can be defined as follows:

$$f: s_i \longrightarrow \eta_i \quad (i = 0, 1, \dots, 2\tau), \quad (4)$$

where $0 \leq \eta_0 < \eta_1 < \dots < \eta_{2\tau} \leq 1$. The linguistic scale function is strictly monotonously increasing function with respect to the subscript of s_i ; in fact, the function value η_i represents the semantics of the linguistic terms.

Now we introduce three kinds of linguistic scale functions as follows:

$$(1) f_1(s_i) = \eta_i = \frac{i}{2\tau} \quad (i = 0, 1, \dots, 2\tau). \quad (5)$$

The evaluation scale of the linguistic information expressed by $f_1(s_i)$ is divided on average.

$$(2) f_2(s_i) = \eta_i = \begin{cases} \frac{c^\tau - c^{t-i}}{2c^\tau - 2}, & i = 0, 1, \dots, \tau; \\ \frac{c^\tau + c^{i-\tau} - 2}{2c^\tau - 2}, & i = \tau + 1, \tau + 2, \dots, 2\tau. \end{cases} \quad (6)$$

For linguistic scale function $f_2(s_i)$, the absolute deviation between adjacent language sets will increase when we extend it from the middle of the given set of language to both ends.

$$(3) f_3(s_i) = \eta_i = \begin{cases} \frac{\tau^{\alpha'} - (\tau - i)^{\alpha'}}{2\tau^{\alpha'}}, & i = 0, 1, \dots, \tau; \\ \frac{\tau^{\beta'} + (i - \tau)^{\beta'}}{2\tau^{\beta'}}, & i = \tau + 1, \tau + 2, \dots, 2\tau. \end{cases} \quad (7)$$

For linguistic scale function $f_3(s_i)$, the absolute deviation between adjacent language sets will decrease when we extend it from the middle of the given linguistic term set to both ends.

The above linguistic scale functions can be developed to $f^* : \bar{S} \rightarrow R^+$ (where R^+ is a nonnegative real number),

which is a continuous and strictly monotonically increasing function.

2.5. The OWA Operator. The ordered weighted averaging (OWA) operator (Yager [24]) is an aggregation operator that includes the minimum, the average, and the maximum as special cases, which is defined as follows.

Definition 5 (Yager [24]). An n dimension OWA operator is a mapping $\text{OWA}: R^n \rightarrow R$ that has an associated weighting vector ω with $\sum_{j=1}^n \omega_j = 1$ ($0 \leq \omega_j \leq 1$), such that

$$\text{OWA}(\gamma_1, \gamma_2, \dots, \gamma_n) = \sum_{j=1}^n \omega_j \gamma'_j, \quad (8)$$

where γ'_j is the j th largest of the arguments $\gamma_1, \gamma_2, \dots, \gamma_n$.

2.6. Cosine Similarity Measure for Fuzzy Sets

Definition 6 (Salton and McGill [25]). Let $X = \{x_1, x_2, \dots, x_n\}$; assume $\xi = \{(x_j, \mu_\xi(x_j)) \mid x_j \in X\}$ and $\sigma = \{(x_j, \mu_\sigma(x_j)) \mid x_j \in X\}$ are two fuzzy sets; the cosine similarity measure between fuzzy sets ξ and σ is defined as follows:

$$C_{\text{FS}}(\xi, \sigma) = \frac{\sum_{j=1}^n \mu_\xi(x_j) \mu_\sigma(x_j)}{\sqrt{\sum_{j=1}^n \mu_\xi^2(x_j)} \sqrt{\sum_{j=1}^n \mu_\sigma^2(x_j)}}. \quad (9)$$

The cosine similarity measure between fuzzy sets ξ and σ satisfies the following properties:

- (1) $0 \leq C_{\text{FS}}(\xi, \sigma) \leq 1$;
- (2) $C_{\text{FS}}(\xi, \sigma) = C_{\text{FS}}(\sigma, \xi)$;
- (3) for $j = 1, 2, \dots, n$ if $\xi = \sigma$, that is, $\mu_\xi(x_j) = \mu_\sigma(x_j)$, then $C_{\text{FS}}(\xi, \sigma) = 1$.

3. Cosine Similarity Measures for Intuitionistic Fuzzy Linguistic Term Sets

3.1. Cosine Similarity Measure for Intuitionistic Fuzzy Linguistic Term Sets. At first, we will present cosine similarity measure for intuitionistic fuzzy linguistic term sets, which includes not only the membership degree and nonmembership degree of the IFLSs but also the linguistic scale function f^* .

Definition 7. Let $\beta = \{(x_j, s_{\beta(x_j)}, \mu_\beta(x_j), \nu_\beta(x_j)) \mid x_j \in X\}$ and $\gamma = \{(x_j, s_{\gamma(x_j)}, \mu_\gamma(x_j), \nu_\gamma(x_j)) \mid x_j \in X\}$ be two IFLSs in $X = \{x_1, x_2, \dots, x_m\}$, and let f^* be a linguistic scale function. Then the cosine similarity measure for intuitionistic fuzzy linguistic term sets between β and γ can be defined as follows:

$$C_{\text{IFLS}}(\beta, \gamma) = \frac{\sum_{j=1}^m [f^*(s_{\beta(x_j)}) \mu_\beta(x_j) f^*(s_{\gamma(x_j)}) \mu_\gamma(x_j) + f^*(s_{\beta(x_j)}) \nu_\beta(x_j) f^*(s_{\gamma(x_j)}) \nu_\gamma(x_j)]}{K \cdot H}, \quad (10)$$

where

K

$$= \sqrt{\sum_{j=1}^m \left[\left(f^* \left(s_{\beta(x_j)} \right) \mu_{\beta} \left(x_j \right) \right)^2 + \left(f^* \left(s_{\beta(x_j)} \right) \nu_{\beta} \left(x_j \right) \right)^2 \right]}, \quad (11)$$

H

$$= \sqrt{\sum_{j=1}^m \left[\left(f^* \left(s_{\gamma(x_j)} \right) \mu_{\gamma} \left(x_j \right) \right)^2 + \left(f^* \left(s_{\gamma(x_j)} \right) \nu_{\gamma} \left(x_j \right) \right)^2 \right]}.$$

The cosine similarity measure between IFLSs β and γ satisfies the following properties:

$$(1) \ 0 \leq C_{IFLS}(\beta, \gamma) \leq 1;$$

$$(2) \ C_{IFLS}(\beta, \gamma) = C_{IFLS}(\gamma, \beta);$$

$$(3) \ \text{for } j = 1, 2, \dots, n \text{ if } \beta = \gamma, \text{ that is, } s_{\beta(x_j)} = s_{\gamma(x_j)}, \mu_{\beta}(x_j) = \mu_{\gamma}(x_j), \text{ and } \nu_{\beta}(x_j) = \nu_{\gamma}(x_j), \text{ then } C_{IFLS}(\beta, \gamma) = 1.$$

Proof. Properties (1), (2), and (3) are obvious; here we omit the proof of property. \square

If we consider the weight of different element $x_j \in X$, now we introduce the intuitionistic fuzzy linguistic weighted cosine similarity measure C_{IFLWS} , which can be defined as follows.

Definition 8. Let $\beta = \{ \langle (x_j, s_{\beta(x_j)}), \mu_{\beta}(x_j), \nu_{\beta}(x_j) \rangle \mid x_j \in X \}$ and $\gamma = \{ \langle (x_j, s_{\gamma(x_j)}), \mu_{\gamma}(x_j), \nu_{\gamma}(x_j) \rangle \mid x_j \in X \}$ be two IFLSs in $X = \{x_1, x_2, \dots, x_m\}$, and let f^* be a linguistic scale function. Then the weighted cosine similarity measure for intuitionistic fuzzy linguistic term sets between β and γ can be defined as follows:

$$C_{IFLWS}(\beta, \gamma) = \frac{\sum_{j=1}^m \omega_j \left[f^* \left(s_{\beta(x_j)} \right) \mu_{\beta} \left(x_j \right) f^* \left(s_{\gamma(x_j)} \right) \mu_{\gamma} \left(x_j \right) + f^* \left(s_{\beta(x_j)} \right) \nu_{\beta} \left(x_j \right) f^* \left(s_{\gamma(x_j)} \right) \nu_{\gamma} \left(x_j \right) \right]}{K_1 \cdot H_1}, \quad (12)$$

where

K_1

$$= \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^* \left(s_{\beta(x_j)} \right) \mu_{\beta} \left(x_j \right) \right)^2 + \left(f^* \left(s_{\beta(x_j)} \right) \nu_{\beta} \left(x_j \right) \right)^2 \right]}, \quad (13)$$

H_1

$$= \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^* \left(s_{\gamma(x_j)} \right) \mu_{\gamma} \left(x_j \right) \right)^2 + \left(f^* \left(s_{\gamma(x_j)} \right) \nu_{\gamma} \left(x_j \right) \right)^2 \right]},$$

ω_j is the weight of $x_j \in X$, and $\sum_{j=1}^m \omega_j = 1$ ($0 \leq \omega_j \leq 1$).

Remark 9. For all $j = 1, 2, \dots, m$, if we take $\omega_j = 1/m$, then the weighted cosine similarity measure $C_{IFLWS}(\beta, \gamma)$ is reduced to the cosine similarity measure $C_{IFLS}(\beta, \gamma)$.

Based on the idea of the OWA operator, we present the intuitionistic fuzzy ordered weighted cosine similarity measure $C_{IFLOWs}(\beta, \gamma)$ between intuitionistic fuzzy linguistic term sets β and γ as follows.

Definition 10. Let $\beta = \{ \langle (x_j, s_{\beta(x_j)}), \mu_{\beta}(x_j), \nu_{\beta}(x_j) \rangle \mid x_j \in X \}$ and $\gamma = \{ \langle (x_j, s_{\gamma(x_j)}), \mu_{\gamma}(x_j), \nu_{\gamma}(x_j) \rangle \mid x_j \in X \}$ be two IFLSs in $X = \{x_1, x_2, \dots, x_m\}$, and let f^* be a linguistic scale function. Then the ordered weighted cosine similarity measure for intuitionistic fuzzy linguistic term sets between β and γ can be defined as follows:

$$C_{IFLOWs}(\beta, \gamma) = \frac{\sum_{j=1}^m \omega_j \left[f^* \left(s_{\beta(x_{\varphi(j)})} \right) \mu_{\beta} \left(x_{\varphi(j)} \right) f^* \left(s_{\gamma(x_{\varphi(j)})} \right) \mu_{\gamma} \left(x_{\varphi(j)} \right) + G \right]}{K_2 \cdot H_2}, \quad (14)$$

where

$$K_2 = \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^* \left(s_{\beta(x_{\varphi(j)})} \right) \mu_{\beta} \left(x_{\varphi(j)} \right) \right)^2 + \left(f^* \left(s_{\beta(x_{\varphi(j)})} \right) \nu_{\beta} \left(x_{\varphi(j)} \right) \right)^2 \right]},$$

$$H_2 = \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^* \left(s_{\gamma(x_{\varphi(j)})} \right) \mu_{\gamma} \left(x_{\varphi(j)} \right) \right)^2 + \left(f^* \left(s_{\gamma(x_{\varphi(j)})} \right) \nu_{\gamma} \left(x_{\varphi(j)} \right) \right)^2 \right]}, \quad (15)$$

$$G = f^* \left(s_{\beta(x_{\varphi(j)})} \right) \nu_{\beta} \left(x_{\varphi(j)} \right) \cdot f^* \left(s_{\gamma(x_{\varphi(j)})} \right) \nu_{\gamma} \left(x_{\varphi(j)} \right),$$

the associated weighting vector $\omega = (\omega_1, \omega_2, \dots, \omega_m)$ with $\sum_{j=1}^m \omega_j = 1$ ($0 \leq \omega_j \leq 1$), and $(\varphi(1), \varphi(2), \dots, \varphi(m))$ is any permutation of $(1, 2, \dots, m)$, such that

$$\begin{aligned} & f^* \left(s_{\beta(x_{\varphi(j)})} \right) \mu_{\beta} \left(x_{\varphi(j)} \right) f^* \left(s_{\gamma(x_{\varphi(j)})} \right) \mu_{\gamma} \left(x_{\varphi(j)} \right) \\ & + f^* \left(s_{\beta(x_{\varphi(j)})} \right) \nu_{\beta} \left(x_{\varphi(j)} \right) f^* \left(s_{\gamma(x_{\varphi(j)})} \right) \nu_{\gamma} \left(x_{\varphi(j)} \right) \\ & \geq f^* \left(s_{\beta(x_{\varphi(j+1)})} \right) \mu_{\beta} \left(x_{\varphi(j+1)} \right) f^* \left(s_{\gamma(x_{\varphi(j+1)})} \right) \\ & \cdot \mu_{\gamma} \left(x_{\varphi(j+1)} \right) + f^* \left(s_{\beta(x_{\varphi(j+1)})} \right) \nu_{\beta} \left(x_{\varphi(j+1)} \right) \\ & \cdot f^* \left(s_{\gamma(x_{\varphi(j+1)})} \right) \nu_{\gamma} \left(x_{\varphi(j+1)} \right). \end{aligned} \quad (16)$$

C_{IFLOWS} is a similarity measure that uses the cosine similarity measure for IFLS in the OWA operator and the linguistic scale function is also applied.

Example 1. Let $S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$ be the linguistic term set, and assume two intuitionistic fuzzy linguistic term sets are $\beta = \{ \langle (x_1, s_4), 1, 0 \rangle, \langle (x_2, s_5), 0.8, 0 \rangle, \langle (x_3, s_3), 0.7, 0.1 \rangle \}$, $\gamma = \{ \langle (x_1, s_5), 0.8, 0.1 \rangle, \langle (x_2, s_5), 1, 0 \rangle, \langle (x_3, s_4), 0.8, 0.1 \rangle \}$. If the linguistic scale function $f^*(s_i) = i/2\tau$ ($\tau = 3$), by (10), we can get

$$C_{\text{IFLS}}(\beta, \gamma) = 0.9867. \quad (17)$$

If $\omega = (\omega_1, \omega_2, \omega_3) = (0.22, 0.4, 0.38)$, then the weighted cosine similarity measure $C_{\text{IFLWS}}(\beta, \gamma) = 0.9889$.

If $\omega = (\omega_1, \omega_2, \omega_3) = (0.22, 0.4, 0.38)$, the ordered weighted cosine similarity measure $C_{\text{IFLOWs}}(\beta, \gamma) = 0.9837$.

3.2. Cosine Similarity Measure for Interval-Valued Intuitionistic Fuzzy Linguistic Term Sets. In the intuitionistic fuzzy linguistic set $\gamma = \{ \langle (x, s_{\gamma(x)}), \mu_{\gamma}(x), \nu_{\gamma}(x) \rangle \mid x \in X \}$, $\mu_{\gamma}(x)$ represents the membership degree of the element x to $s_{\gamma(x)}$, and $\nu_{\gamma}(x)$ represents the nonmembership degree of the element x to $s_{\gamma(x)}$; they are all precise values in $[0, 1]$. But in some circumstances, it is difficult to provide the precise membership degree and nonmembership degree of the element x to $s_{\gamma(x)}$. Atanassov and Gargov [26, 27] proposed the interval-valued intuitionistic fuzzy linguistic term set (IVIFLS), and the definition of interval-valued intuitionistic fuzzy linguistic term set is given as follows.

Definition 11. Let $X = \{x_1, x_2, \dots, x_n\}$ be a fixed set and $s_{\tilde{\gamma}(x)}$ $\in \bar{S}$ an interval-valued intuitionistic fuzzy linguistic term set $\tilde{\gamma}$ in X is defined as

$$\tilde{\gamma} = \{ \langle (x, s_{\tilde{\gamma}(x)}), \mu_{\tilde{\gamma}}(x), \nu_{\tilde{\gamma}}(x) \rangle \mid x \in X \}, \quad (18)$$

where $\mu_{\tilde{\gamma}}(x) = [\mu_{\tilde{\gamma}l}(x), \mu_{\tilde{\gamma}r}(x)] \subset [0, 1]$ and $\nu_{\tilde{\gamma}}(x) = [\nu_{\tilde{\gamma}l}(x), \nu_{\tilde{\gamma}r}(x)] \subset [0, 1]$ are the interval membership degree and the interval nonmembership degree of the element x to $s_{\tilde{\gamma}(x)}$, respectively, and $0 \leq \mu_{\tilde{\gamma}r}(x) + \nu_{\tilde{\gamma}r}(x) \leq 1$ ($x \in X$).

Based on the cosine similarity measure for intuitionistic fuzzy linguistic term sets β and γ , we present the cosine similarity measure for interval-valued intuitionistic fuzzy linguistic term sets $\tilde{\beta}$ and $\tilde{\gamma}$ as follows.

Definition 12. Let $\tilde{\beta} = \{ \langle (x_j, s_{\tilde{\beta}(x_j)}), \mu_{\tilde{\beta}}(x_j), \nu_{\tilde{\beta}}(x_j) \rangle \mid x_j \in X \}$ and $\tilde{\gamma} = \{ \langle (x_j, s_{\tilde{\gamma}(x_j)}), \mu_{\tilde{\gamma}}(x_j), \nu_{\tilde{\gamma}}(x_j) \rangle \mid x_j \in X \}$ be two IVIFLSs in $X = \{x_1, x_2, \dots, x_m\}$, and let f^* be a linguistic scale function. Then the cosine similarity measure for IVIFLSs between $\tilde{\beta}$ and $\tilde{\gamma}$ can be defined as follows:

$$C_{\text{IVIFLS}}(\tilde{\beta}, \tilde{\gamma}) = \frac{\sum_{j=1}^m \left[f^* \left(s_{\tilde{\beta}(x_j)} \right) \mu_{\tilde{\beta}l}(x_j) f^* \left(s_{\tilde{\gamma}(x_j)} \right) \mu_{\tilde{\gamma}l}(x_j) + f^* \left(s_{\tilde{\beta}(x_j)} \right) \nu_{\tilde{\beta}l}(x_j) f^* \left(s_{\tilde{\gamma}(x_j)} \right) \nu_{\tilde{\gamma}l}(x_j) + U \right]}{Q \cdot R}, \quad (19)$$

where

$$U = f^* \left(s_{\tilde{\beta}(x_j)} \right) \mu_{\tilde{\beta}r}(x_j) f^* \left(s_{\tilde{\gamma}(x_j)} \right) \mu_{\tilde{\gamma}r}(x_j) + f^* \left(s_{\tilde{\beta}(x_j)} \right) \nu_{\tilde{\beta}r}(x_j) f^* \left(s_{\tilde{\gamma}(x_j)} \right) \nu_{\tilde{\gamma}r}(x_j),$$

$$Q = \sqrt{\sum_{j=1}^m \left[\left(f^* \left(s_{\tilde{\beta}(x_j)} \right) \mu_{\tilde{\beta}l}(x_j) \right)^2 + \left(f^* \left(s_{\tilde{\beta}(x_j)} \right) \mu_{\tilde{\beta}r}(x_j) \right)^2 + \left(f^* \left(s_{\tilde{\beta}(x_j)} \right) \nu_{\tilde{\beta}l}(x_j) \right)^2 + \left(f^* \left(s_{\tilde{\beta}(x_j)} \right) \nu_{\tilde{\beta}r}(x_j) \right)^2 \right]}, \quad (20)$$

$$R = \sqrt{\sum_{j=1}^m \left[\left(f^* \left(s_{\tilde{\gamma}(x_j)} \right) \mu_{\tilde{\gamma}l}(x_j) \right)^2 + \left(f^* \left(s_{\tilde{\gamma}(x_j)} \right) \mu_{\tilde{\gamma}r}(x_j) \right)^2 + \left(f^* \left(s_{\tilde{\gamma}(x_j)} \right) \nu_{\tilde{\gamma}l}(x_j) \right)^2 + \left(f^* \left(s_{\tilde{\gamma}(x_j)} \right) \nu_{\tilde{\gamma}r}(x_j) \right)^2 \right]}.$$

The cosine similarity measure between IVIFLSs $\tilde{\beta}$ and $\tilde{\gamma}$ also satisfies the following properties:

- (1) $0 \leq C_{\text{IVIFLS}}(\tilde{\beta}, \tilde{\gamma}) \leq 1$;
- (2) $C_{\text{IVIFLS}}(\tilde{\beta}, \tilde{\gamma}) = C_{\text{IVIFLS}}(\tilde{\gamma}, \tilde{\beta})$;
- (3) for $j = 1, 2, \dots, n$ if $\tilde{\beta} = \tilde{\gamma}$, that is, $s_{\tilde{\beta}(x_j)} = s_{\tilde{\gamma}(x_j)}$, $\mu_{\tilde{\beta}l}(x_j) = \mu_{\tilde{\gamma}l}(x_j)$, $\mu_{\tilde{\beta}r}(x_j) = \mu_{\tilde{\gamma}r}(x_j)$, $\nu_{\tilde{\beta}l}(x_j) = \nu_{\tilde{\gamma}l}(x_j)$ and $\nu_{\tilde{\beta}r}(x_j) = \nu_{\tilde{\gamma}r}(x_j)$, then $C_{\text{IVIFLS}}(\tilde{\beta}, \tilde{\gamma}) = 1$.

Next we go on studying the weighted cosine similarity measure between IVIFLSs; it can be defined as follows.

Definition 13. Let $\tilde{\beta} = \{ \langle (x_j, s_{\tilde{\beta}(x_j)}), \mu_{\tilde{\beta}}(x_j), \nu_{\tilde{\beta}}(x_j) \rangle \mid x_j \in X \}$ and $\tilde{\gamma} = \{ \langle (x_j, s_{\tilde{\gamma}(x_j)}), \mu_{\tilde{\gamma}}(x_j), \nu_{\tilde{\gamma}}(x_j) \rangle \mid x_j \in X \}$ be two IVIFLSs in $X = \{x_1, x_2, \dots, x_m\}$, ω_j is the weight of $x_j \in X$, and $\sum_{j=1}^m \omega_j = 1$ ($0 \leq \omega_j \leq 1$). Assume f^* be a linguistic scale function, then the weighted cosine similarity measure between IVIFLSs $\tilde{\beta}$ and $\tilde{\gamma}$ can be defined as follows:

$$C_{\text{IVIFLWS}}(\tilde{\beta}, \tilde{\gamma}) = \frac{\sum_{j=1}^m \omega_j \left[f^*(s_{\tilde{\beta}(x_j)}) \mu_{\tilde{\beta}l}(x_j) f^*(s_{\tilde{\gamma}(x_j)}) \mu_{\tilde{\gamma}l}(x_j) + f^*(s_{\tilde{\beta}(x_j)}) \nu_{\tilde{\beta}l}(x_j) f^*(s_{\tilde{\gamma}(x_j)}) \nu_{\tilde{\gamma}l}(x_j) + U \right]}{Q_1 \cdot R_1}, \quad (21)$$

where

$$U = f^*(s_{\tilde{\beta}(x_j)}) \mu_{\tilde{\beta}r}(x_j) f^*(s_{\tilde{\gamma}(x_j)}) \mu_{\tilde{\gamma}r}(x_j) + f^*(s_{\tilde{\beta}(x_j)}) \nu_{\tilde{\beta}r}(x_j) f^*(s_{\tilde{\gamma}(x_j)}) \nu_{\tilde{\gamma}r}(x_j),$$

$$Q_1 = \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^*(s_{\tilde{\beta}(x_j)}) \mu_{\tilde{\beta}l}(x_j) \right)^2 + \left(f^*(s_{\tilde{\beta}(x_j)}) \mu_{\tilde{\beta}r}(x_j) \right)^2 + \left(f^*(s_{\tilde{\beta}(x_j)}) \nu_{\tilde{\beta}l}(x_j) \right)^2 + \left(f^*(s_{\tilde{\beta}(x_j)}) \nu_{\tilde{\beta}r}(x_j) \right)^2 \right]}, \quad (22)$$

$$R_1 = \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^*(s_{\tilde{\gamma}(x_j)}) \mu_{\tilde{\gamma}l}(x_j) \right)^2 + \left(f^*(s_{\tilde{\gamma}(x_j)}) \mu_{\tilde{\gamma}r}(x_j) \right)^2 + \left(f^*(s_{\tilde{\gamma}(x_j)}) \nu_{\tilde{\gamma}l}(x_j) \right)^2 + \left(f^*(s_{\tilde{\gamma}(x_j)}) \nu_{\tilde{\gamma}r}(x_j) \right)^2 \right]}.$$

Remark 14. For all $j = 1, 2, \dots, m$, if we take $\omega_j = 1/m$, then the weighted cosine similarity measure $C_{\text{IVIFLWS}}(\tilde{\beta}, \tilde{\gamma})$ is reduced to the cosine similarity measure $C_{\text{IVIFLS}}(\tilde{\beta}, \tilde{\gamma})$.

Remark 15. For all $j = 1, 2, \dots, m$, if $\mu_{\tilde{\beta}l}(x_j) = \mu_{\tilde{\beta}r}(x_j)$, $\mu_{\tilde{\gamma}l}(x_j) = \mu_{\tilde{\gamma}r}(x_j)$, then the weighted cosine similarity measure $C_{\text{IVIFLWS}}(\tilde{\beta}, \tilde{\gamma})$ is reduced to the weighted cosine similarity measure $C_{\text{IFLWS}}(\beta, \gamma)$.

Remark 16. For all $j = 1, 2, \dots, m$, if $\omega_j = 1/m$ and $\mu_{\tilde{\beta}l}(x_j) = \mu_{\tilde{\beta}r}(x_j)$, $\mu_{\tilde{\gamma}l}(x_j) = \mu_{\tilde{\gamma}r}(x_j)$, then the weighted cosine similarity measure $C_{\text{IVIFLWS}}(\tilde{\beta}, \tilde{\gamma})$ is reduced to the cosine similarity measure $C_{\text{IFLS}}(\beta, \gamma)$ for IFLSs.

Similarly, we also apply the OWA operator and the cosine similarity measure for IVIFLS to present the interval-valued intuitionistic fuzzy ordered weighted cosine similarity measure $C_{\text{IVIFLWS}}(\tilde{\beta}, \tilde{\gamma})$ between the interval-valued intuitionistic fuzzy linguistic term sets $\tilde{\beta}$ and $\tilde{\gamma}$ as follows.

Definition 17. Let $\tilde{\beta} = \{ \langle (x_j, s_{\tilde{\beta}(x_j)}), \mu_{\tilde{\beta}}(x_j), \nu_{\tilde{\beta}}(x_j) \rangle \mid x_j \in X \}$ and $\tilde{\gamma} = \{ \langle (x_j, s_{\tilde{\gamma}(x_j)}), \mu_{\tilde{\gamma}}(x_j), \nu_{\tilde{\gamma}}(x_j) \rangle \mid x_j \in X \}$ be two IVIFLSs in $X = \{x_1, x_2, \dots, x_m\}$, ω_j is the weight of $x_j \in X$, and $\sum_{j=1}^m \omega_j = 1$ ($0 \leq \omega_j \leq 1$). Assume f^* be a linguistic scale function, then the ordered weighted cosine similarity measure between IVIFLSs $\tilde{\beta}$ and $\tilde{\gamma}$ can be defined as follows:

$$C_{\text{IVIFLWS}}(\tilde{\beta}, \tilde{\gamma}) = \frac{\sum_{j=1}^m \omega_j \left[f^*(s_{\tilde{\beta}(x_{\varphi(j)})}) f^*(s_{\tilde{\gamma}(x_{\varphi(j)})}) (\mu_{\tilde{\beta}l}(x_{\varphi(j)}) \mu_{\tilde{\gamma}l}(x_{\varphi(j)}) + \nu_{\tilde{\beta}l}(x_{\varphi(j)}) \nu_{\tilde{\gamma}l}(x_{\varphi(j)})) + U_2 \right]}{Q_2 \cdot R_2}, \quad (23)$$

where

$$U_2 = f^*(s_{\tilde{\beta}(x_{\varphi(j)})}) \mu_{\tilde{\beta}r}(x_{\varphi(j)}) f^*(s_{\tilde{\gamma}(x_{\varphi(j)})}) \mu_{\tilde{\gamma}r}(x_{\varphi(j)}) + f^*(s_{\tilde{\beta}(x_{\varphi(j)})}) \nu_{\tilde{\beta}r}(x_{\varphi(j)}) f^*(s_{\tilde{\gamma}(x_{\varphi(j)})}) \nu_{\tilde{\gamma}r}(x_{\varphi(j)}),$$

$$\begin{aligned}
Q_2 &= \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^* \left(s_{\tilde{\beta}(x_{\varphi(j)})} \right) \right)^2 \left(\left(\mu_{\tilde{\beta}l}(x_{\varphi(j)}) \right)^2 + \mu_{\tilde{\beta}r}(x_{\varphi(j)}) \right)^2 \right] + \left(f^* \left(s_{\tilde{\beta}(x_{\varphi(j)})} \right) \right)^2 \left(\left(\nu_{\tilde{\beta}l}(x_{\varphi(j)}) \right)^2 + \nu_{\tilde{\beta}r}(x_{\varphi(j)}) \right)^2 \right]}, \\
R_2 &= \sqrt{\sum_{j=1}^m \omega_j \left[\left(f^* \left(s_{\tilde{\gamma}(x_{\varphi(j)})} \right) \right)^2 \left(\left(\mu_{\tilde{\gamma}l}(x_{\varphi(j)}) \right)^2 + \mu_{\tilde{\gamma}r}(x_{\varphi(j)}) \right)^2 \right] + \left(f^* \left(s_{\tilde{\gamma}(x_{\varphi(j)})} \right) \right)^2 \left(\left(\nu_{\tilde{\gamma}l}(x_{\varphi(j)}) \right)^2 + \nu_{\tilde{\gamma}r}(x_{\varphi(j)}) \right)^2 \right]},
\end{aligned} \tag{24}$$

and $(\varphi(1), \varphi(2), \dots, \varphi(m))$ is any permutation of $(1, 2, \dots, m)$, such that

$$\begin{aligned}
& f^* \left(s_{\tilde{\beta}(x_{\varphi(j)})} \right) f^* \left(s_{\tilde{\gamma}(x_{\varphi(j)})} \right) \left[\mu_{\tilde{\beta}l}(x_{\varphi(j)}) \mu_{\tilde{\gamma}l}(x_{\varphi(j)}) \right. \\
& + \nu_{\tilde{\beta}l}(x_{\varphi(j)}) \nu_{\tilde{\gamma}l}(x_{\varphi(j)}) + \mu_{\tilde{\beta}r}(x_{\varphi(j)}) \mu_{\tilde{\gamma}r}(x_{\varphi(j)}) \\
& + \left. \nu_{\tilde{\beta}r}(x_{\varphi(j)}) \nu_{\tilde{\gamma}r}(x_{\varphi(j)}) \right] \geq f^* \left(s_{\tilde{\beta}(x_{\varphi(j+1)})} \right) \\
& \cdot f^* \left(s_{\tilde{\gamma}(x_{\varphi(j+1)})} \right) \left[\mu_{\tilde{\beta}l}(x_{\varphi(j+1)}) \mu_{\tilde{\gamma}l}(x_{\varphi(j+1)}) \right. \\
& + \nu_{\tilde{\beta}l}(x_{\varphi(j+1)}) \nu_{\tilde{\gamma}l}(x_{\varphi(j+1)}) \\
& + \left. \mu_{\tilde{\beta}r}(x_{\varphi(j+1)}) \mu_{\tilde{\gamma}r}(x_{\varphi(j+1)}) \right. \\
& + \left. \nu_{\tilde{\beta}r}(x_{\varphi(j+1)}) \nu_{\tilde{\gamma}r}(x_{\varphi(j+1)}) \right].
\end{aligned} \tag{25}$$

4. Applications of the Cosine Similarity Measure

In this section, we will apply the cosine similarity measures of IFLSs and IVIFLSs to pattern recognition and medical diagnosis.

4.1. Intuitionistic Fuzzy Cosine Similarity Measure for Pattern Recognition

Example 2. Let $X = \{x_1, x_2, x_3, x_4\}$, the linguistic term set $S = \{s_0 = \text{very poor}, s_1 = \text{poor}, s_2 = \text{slightly poor}, s_3 = \text{fair}, s_4 = \text{slightly good}, s_5 = \text{good}, s_6 = \text{very good}\}$. We consider some known patterns Q_1, Q_2, Q_3, Q_4 , which are represented by the IFLSs as follows:

$$\begin{aligned}
Q_1 &= \{ \langle (x_1, s_4), 0.8, 0.1 \rangle, \langle (x_2, s_3), 1, 0 \rangle, \\
& \quad \langle (x_3, s_5), 0.7, 0.1 \rangle, \langle (x_4, s_3), 0.6, 0.3 \rangle \}, \\
Q_2 &= \{ \langle (x_1, s_3), 0.9, 0 \rangle, \langle (x_2, s_3), 0.8, 0.1 \rangle, \\
& \quad \langle (x_3, s_4), 0.9, 0.1 \rangle, \langle (x_4, s_3), 0.6, 0.3 \rangle \}, \\
Q_3 &= \{ \langle (x_1, s_4), 0.7, 0.1 \rangle, \langle (x_2, s_4), 0.9, 0 \rangle, \\
& \quad \langle (x_3, s_5), 0.8, 0.1 \rangle, \langle (x_4, s_3), 0.8, 0.1 \rangle \}, \\
Q_4 &= \{ \langle (x_1, s_5), 0.6, 0.2 \rangle, \langle (x_2, s_4), 0.7, 0.2 \rangle, \\
& \quad \langle (x_3, s_5), 0.8, 0.2 \rangle, \langle (x_4, s_2), 0.7, 0.2 \rangle \}.
\end{aligned} \tag{26}$$

If an unknown pattern $Q = \{ \langle (x_1, s_4), 0.8, 0.2 \rangle, \langle (x_2, s_4), 1, 0 \rangle, \langle (x_3, s_4), 0.8, 0.1 \rangle, \langle (x_4, s_3), 0.8, 0.2 \rangle \}$, in order to classify the pattern Q in Q_1, Q_2, Q_3, Q_4 , we can calculate the weighted cosine similarity measure between Q and Q_i ($i = 1, 2, 3, 4$), respectively. The best Q_i is derived by $R = \arg \max_{1 \leq i \leq 4} \{ C_{IFLWS}(Q_i, Q) \}$. Assume the weight of (x_1, x_2, x_3, x_4) is $(0.22, 0.3, 0.2, 0.28)$ and let the linguistic scale function $f^* = f_1(s_i) = i/2\tau$ ($\tau = 3$), and by applying (10), (12), and (14), we obtain the cosine similarity measures between Q and Q_i ($i = 1, 2, 3, 4$), and the results are shown in Table 1.

From the result shown in Table 1, we can conclude that the pattern Q belongs to the pattern Q_3 .

To illustrate the influence of the linguistic scale function f^* on decision-making, we utilize the different linguistic scaling functions in the proposed cosine similarity measures.

Let

$$\begin{aligned}
f^* &= f_2(s_i) \\
&= \begin{cases} \frac{c^\tau - c^{\tau-i}}{2c^\tau - 2}, & i = 0, 1, \dots, \tau, \\ \frac{c^\tau + c^{i-\tau} - 2}{2c^\tau - 2}, & i = \tau + 1, \tau + 2, \dots, 2\tau, \end{cases}
\end{aligned} \tag{27}$$

and $c = 1.4, \tau = 3$, then the results are listed in Table 2.

If

$$f_3^*(s_i) = \begin{cases} \frac{\tau^{\alpha'} - (\tau - i)^{\alpha'}}{2\tau^{\alpha'}}, & i = 0, 1, \dots, \tau, \\ \frac{\tau^{\beta'} + (i - \tau)^{\beta'}}{2\tau^{\beta'}}, & i = \tau + 1, \tau + 2, \dots, 2\tau, \end{cases} \tag{28}$$

and $\tau = 3, \alpha' = \beta' = 0.8$, the results of the cosine similarity measure are shown in Table 3.

As we can see from Tables 2 and 3, we know that the pattern Q should be classified in Q_3 in most cases. It is a little different when the linguistic scale function $f^* = f_2(s_i)$ (the linguistic scale function which can be considered as the actual semantic situation), then the decision-makers can select the appropriate linguistic scale function f^* according to their interests.

4.2. Interval-Valued Intuitionistic Fuzzy Cosine Similarity Measure for Medical Diagnosis. In this subsection, we will utilize the interval-valued intuitionistic fuzzy cosine similarity measure to discuss the medical diagnosis. In fact, it is also a pattern recognition problem.

TABLE 1: Intuitionistic fuzzy linguistic cosine similarity measures for $f^* = f_1(s_i)$.

	Q_1	Q_2	Q_3	Q_4
$C_{IFLS}(Q, Q_i)$	0.9828	0.9589	0.9859	0.9476
$C_{IFLWS}(Q, Q_i)$	0.9826	0.9587	0.9853	0.9488
$C_{IFLWS}(Q, Q_i)$	0.9824	0.9582	0.9849	0.9473

TABLE 2: Intuitionistic fuzzy linguistic cosine similarity measures for $f^* = f_2(s_i)$.

	Q_1	Q_2	Q_3	Q_4
$C_{IFLS}(Q, Q_i)$	0.9861	0.9655	0.9854	0.9600
$C_{IFLWS}(Q, Q_i)$	0.9862	0.9654	0.9871	0.9598
$C_{IFLWS}(Q, Q_i)$	0.9853	0.9849	0.9848	0.9713

TABLE 3: Intuitionistic fuzzy linguistic cosine similarity measures for $f^* = f_3(s_i)$.

	Q_1	Q_2	Q_3	Q_4
$C_{IFLS}(Q, Q_i)$	0.9806	0.9530	0.9875	0.9494
$C_{IFLWS}(Q, Q_i)$	0.9800	0.9526	0.9889	0.9472
$C_{IFLWS}(Q, Q_i)$	0.9805	0.9519	0.9870	0.9458

Example 3. Let us consider that the doctor makes a diagnosis $D = \{D_1(\text{viral fever}), D_2(\text{typhoid}), D_3(\text{pneumonia}), D_4(\text{stomach problem})\}$, and assume that a set of symptoms $X = \{x_1(\text{fever}), x_2(\text{cough}), x_3(\text{headache}), x_4(\text{stomach pain})\}$, and each symptom can be presented as the linguistic term set $S_1 = \{s_0 = \text{very low}, s_1 = \text{low}, s_2 = \text{slightly low}, s_3 = \text{normal}, s_4 = \text{slightly high}, s_5 = \text{high}, s_6 = \text{very high}\}$, $S_j = \{s_0 = \text{none}, s_1 = \text{very slight}, s_2 = \text{slight}, s_3 = \text{a little terrible}, s_4 = \text{terrible}, s_5 = \text{very terrible}, s_6 = \text{insufferable}\}$ ($j = 2, 3, 4$). Suppose that the patient P has all the symptoms, which is represented by the IVIFLS as follows:

$$\begin{aligned}
 P(\text{patient}) = & \{ \langle (x_1, s_3), [0.4, 0.5], [0.2, 0.3] \rangle, \\
 & \langle (x_2, s_4), [0.7, 0.8], [0.1, 0.2] \rangle, \\
 & \langle (x_3, s_3), [0.9, 1], [0, 0.1] \rangle, \\
 & \langle (x_4, s_1), [0.3, 0.5], [0.2, 0.4] \rangle \}.
 \end{aligned} \tag{29}$$

Each symptom diagnosis D_i ($i = 1, 2, 3, 4$) can be also represented as IVIFLSs as follows:

$$\begin{aligned}
 D_1 = & \{ \langle (x_1, s_4), [0.8, 0.9], [0, 0.1] \rangle, \\
 & \langle (x_2, s_3), [0.7, 0.9], [0.1, 0.2] \rangle, \\
 & \langle (x_3, s_4), [0.5, 0.6], [0.2, 0.3] \rangle, \\
 & \langle (x_4, s_0), [0.7, 0.9], [0.1, 0.2] \rangle \}, \\
 D_2 = & \{ \langle (x_1, s_3), [0.5, 0.6], [0.1, 0.3] \rangle, \\
 & \langle (x_2, s_4), [0.8, 0.9], [0, 0.1] \rangle,
 \end{aligned}$$

TABLE 4: Interval intuitionistic fuzzy linguistic cosine similarity measures for $f^* = f_1(s_i)$.

	D_1	D_2	D_3	D_4
$C_{IVIFLS}(P, D_i)$	0.8685	0.9004	0.7748	0.5481
$C_{IVIFLWS}(P, D_i)$	0.8809	0.9119	0.8619	0.6102
$C_{IVIFLWS}(P, D_i)$	0.8967	0.9267	0.8212	0.5310

TABLE 5: Interval intuitionistic fuzzy linguistic cosine similarity measures for $f^* = f_2(s_i)$.

	D_1	D_2	D_3	D_4
$C_{IVIFLS}(P, D_i)$	0.8842	0.8966	0.7865	0.6839
$C_{IVIFLWS}(P, D_i)$	0.9003	0.9139	0.8657	0.7354
$C_{IVIFLWS}(P, D_i)$	0.9051	0.9247	0.8292	0.6726

TABLE 6: Interval intuitionistic fuzzy linguistic cosine similarity measures for $f^* = f_3(s_i)$.

	D_1	D_2	D_3	D_4
$C_{IVIFLS}(P, D_i)$	0.8550	0.9044	0.7724	0.4681
$C_{IVIFLWS}(P, D_i)$	0.8641	0.9247	0.8616	0.5297
$C_{IVIFLWS}(P, D_i)$	0.8880	0.9292	0.8181	0.4486

$$\begin{aligned}
 & \langle (x_3, s_3), [0.6, 0.8], [0.1, 0.2] \rangle, \\
 & \langle (x_4, s_2), [0.4, 0.6], [0.1, 0.2] \rangle \}, \\
 D_3 = & \{ \langle (x_1, s_5), [0.7, 0.8], [0.1, 0.2] \rangle, \\
 & \langle (x_2, s_5), [0.7, 0.9], [0, 0.1] \rangle, \\
 & \langle (x_3, s_1), [0.4, 0.6], [0.2, 0.4] \rangle, \\
 & \langle (x_4, s_1), [0.3, 0.5], [0.2, 0.4] \rangle \}, \\
 D_4 = & \{ \langle (x_1, s_1), [0.8, 0.9], [0, 0.1] \rangle, \\
 & \langle (x_2, s_2), [0.7, 0.8], [0.1, 0.2] \rangle, \\
 & \langle (x_3, s_1), [0.7, 0.9], [0, 0.1] \rangle, \\
 & \langle (x_4, s_5), [0.8, 0.9], [0, 0.1] \rangle \}
 \end{aligned} \tag{30}$$

We assume that the weight vector of (x_1, x_2, x_3, x_4) is $(0.25, 0.4, 0.15, 0.2)$ and let the linguistic scale function $f^* = f_1(s_i) = i/2\tau$ ($\tau = 3$), and by applying (19), (21), and (23), we obtain the cosine similarity measures between P and D_i ($i = 1, 2, 3, 4$), and the results are shown in Table 4.

From the results of Table 4, we can see that the diagnosis of the patient P is D_2 (typhoid).

To illustrate the influence of the linguistic scale function f^* on decision-making, we utilize the different linguistic scaling functions $f_2(s_i)$ and $f_3(s_i)$ (the parameters are the same as Example 2) in the cosine similarity measures between IVIFLSs, and the results are shown in Tables 5 and 6.

TABLE 7: Cosine similarity measures for intuitionistic fuzzy sets.

	Q'_1	Q'_2	Q'_3	Q'_4
$C_{IFS}(Q', Q'_i)$	0.9902	0.9733	0.9954	0.9772
$C_{IFWS}(Q', Q'_i)$	0.9898	0.9734	0.9956	0.9761
$C_{IFOWS}(Q', Q'_i)$	0.9890	0.9742	0.9949	0.9773

In these cases, we still find that the diagnosis of the patient P should be classified in D_2 (typhoid).

4.3. Comparison Analysis with Existing Cosine Similarity Measure. To illustrate the validity and advantage of the proposed cosine similarity measure for pattern recognition and medical diagnosis, we now use the existing cosine similarity measure between intuitionistic fuzzy sets and interval-valued intuitionistic fuzzy sets (Ye [20]) for comparison analysis. Here we consider the intuitionistic fuzzy set in Example 2 and interval-valued intuitionistic fuzzy set in Example 3, respectively.

Example 2'. Let $X = \{x_1, x_2, x_3, x_4\}$; we consider some known patterns Q'_1, Q'_2, Q'_3, Q'_4 , which are represented by the IFSs as follows:

$$\begin{aligned}
Q'_1 &= \{\langle x_1, 0.8, 0.1 \rangle, \langle x_2, 1, 0 \rangle, \langle x_3, 0.7, 0.1 \rangle, \\
&\quad \langle x_4, 0.6, 0.3 \rangle\}, \\
Q'_2 &= \{\langle x_1, 0.9, 0 \rangle, \langle x_2, 0.8, 0.1 \rangle, \langle x_3, 0.9, 0.1 \rangle, \\
&\quad \langle x_4, 0.6, 0.3 \rangle\}, \\
Q'_3 &= \{\langle x_1, 0.7, 0.1 \rangle, \langle x_2, 0.9, 0 \rangle, \langle x_3, 0.8, 0.1 \rangle, \\
&\quad \langle x_4, 0.8, 0.1 \rangle\}, \\
Q'_4 &= \{\langle x_1, 0.6, 0.2 \rangle, \langle x_2, 0.7, 0.2 \rangle, \langle x_3, 0.8, 0.2 \rangle, \\
&\quad \langle x_4, 0.7, 0.2 \rangle\}.
\end{aligned} \tag{31}$$

The unknown pattern $Q' = \{\langle x_1, 0.8, 0.2 \rangle, \langle x_2, 1, 0 \rangle, \langle x_3, 0.8, 0.1 \rangle, \langle x_4, 0.8, 0.2 \rangle\}$ and the weight of (x_1, x_2, x_3, x_4) is still $(0.22, 0.3, 0.2, 0.28)$. We can calculate the intuitionistic fuzzy cosine similarity measure C_{IFS} , the intuitionistic fuzzy weighted cosine similarity measure C_{IFWS} , and the intuitionistic fuzzy ordered weighted cosine similarity measure C_{IFOWS} between Q' and Q'_i ($i = 1, 2, 3, 4$), respectively, and the results are shown in Table 7.

The results of Table 7 show that the pattern Q' can be classified in Q'_3 ; this shows the effectiveness of the proposed cosine similarity measures in this paper.

Example 3'. Let us consider that the doctor makes a diagnosis $D = \{D'_1(\text{viral fever}), D'_2(\text{typhoid}), D'_3(\text{pneumonia}), D'_4(\text{stomach problem})\}$, and assume that a set of symptoms $X = \{x_1(\text{fever}), x_2(\text{cough}), x_3(\text{headache}),$

TABLE 8: Cosine similarity measures for interval-valued intuitionistic fuzzy sets.

	D'_1	D'_2	D'_3	D'_4
$C_{IFS}(P', D'_i)$	0.9173	0.9617	0.8757	0.9546
$C_{IFWS}(P', D'_i)$	0.9341	0.9638	0.9019	0.9566
$C_{IFOWS}(P', D'_i)$	0.9423	0.9650	0.8752	0.9614

$x_4(\text{stomach pain})\}$, and each symptom can be represented as the interval-valued intuitionistic fuzzy set as follows:

$$\begin{aligned}
D'_1 &= \{\langle x_1, [0.8, 0.9], [0, 0.1] \rangle, \\
&\quad \langle x_2, [0.7, 0.9], [0.1, 0.2] \rangle, \langle x_3, [0.5, 0.6], [0.2, 0.3] \rangle, \\
&\quad \langle x_4, [0.7, 0.9], [0.1, 0.2] \rangle\}, \\
D'_2 &= \{\langle x_1, [0.5, 0.6], [0.1, 0.3] \rangle, \\
&\quad \langle x_2, [0.8, 0.9], [0, 0.1] \rangle, \langle x_3, [0.6, 0.8], [0.1, 0.2] \rangle, \\
&\quad \langle x_4, [0.4, 0.6], [0.1, 0.2] \rangle\}, \\
D'_3 &= \{\langle x_1, [0.7, 0.8], [0.1, 0.2] \rangle, \\
&\quad \langle x_2, [0.7, 0.9], [0, 0.1] \rangle, \langle x_3, [0.4, 0.6], [0.2, 0.4] \rangle, \\
&\quad \langle x_4, [0.3, 0.5], [0.2, 0.4] \rangle\}, \\
D'_4 &= \{\langle x_1, [0.8, 0.9], [0, 0.1] \rangle, \\
&\quad \langle x_2, [0.7, 0.8], [0.1, 0.2] \rangle, \langle x_3, [0.7, 0.9], [0, 0.1] \rangle, \\
&\quad \langle x_4, [0.8, 0.9], [0, 0.1] \rangle\}.
\end{aligned} \tag{32}$$

Suppose that the patient P' has all the symptoms, which is represented by the following interval-valued intuitionistic fuzzy set:

$$\begin{aligned}
P'(\text{patient}) &= \{\langle x_1, [0.4, 0.5], [0.2, 0.3] \rangle, \\
&\quad \langle x_2, [0.7, 0.8], [0.1, 0.2] \rangle, \langle x_3, [0.9, 1], [0, 0.1] \rangle, \\
&\quad \langle x_4, [0.3, 0.5], [0.2, 0.4] \rangle\}.
\end{aligned} \tag{33}$$

The weight vector of (x_1, x_2, x_3, x_4) is $(0.25, 0.4, 0.15, 0.2)$. Using the cosine similarity measure between IVIFSs in [22], we can calculate the interval-valued intuitionistic fuzzy cosine similarity measure C_{IVIFS} , the interval-valued intuitionistic fuzzy weighted cosine similarity measure C_{IVIFWS} , and the interval-valued intuitionistic fuzzy ordered weighted cosine similarity measure $C_{IVIFOWS}$ between P' and D'_i ($i = 1, 2, 3, 4$), respectively, and the results are shown in Table 8.

As we can see from Table 8, the patient P' is assigned to the diagnosis D'_2 (typhoid), and the result is the same as the method that we presented in this paper. According to the comparative analysis of Sections 4.1 and 4.2, the cosine similarity measures in this paper have the following advantage. It is reasonable that the proposed cosine similarity measures between IFLSs and IVIFLSs are defined based on linguistic scale function f^* , the decision-makers can flexibly select the linguistic scale function f^* depending on

their preferences and the actual semantic situations, and the decision-maker can employ it to address practical decision-making problems with precision.

5. Conclusions

In this paper, we propose the cosine similarity measures for IFLSs and IVIFLSs, which are expressed by the cosine function based on linguistic scale function. Then, the weighted cosine similarity measure and the ordered weighted cosine similarity measure for intuitionistic fuzzy linguistic term sets are introduced. The main advantage of the proposed cosine similarity measures is that the decision-makers can flexibly select the linguistic scale function depending on the actual semantic situation. Furthermore, we present the application of the cosine similarity measures for intuitionistic fuzzy linguistic term sets and interval-valued intuitionistic fuzzy linguistic term sets to pattern recognition and medical diagnosis, and the existing cosine similarity measures are compared with the proposed cosine similarity measures by the illustrative example. In future research, the developed cosine similarity measures will be extended to the intuitionistic fuzzy uncertain linguistic sets and it can be applied in other related decision-making.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is fully supported by a grant by National Natural Science Foundation of Hunan (2015JJ6041), by National Natural Science Foundation of China (11501191), by National Social Science Foundation of China (15BTJ028), by Major Projects of the National Social Science Foundation of China (17ZDA046), and by the Key International Collaboration Project of the National Nature Science Foundation of China (no. 71210003, Research on Electronic Business Based on the Users Behavior).

References

- [1] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [2] R. R. Yager, "Multiple objective decision-making using fuzzy sets," *International Journal of Man-Machine Studies*, vol. 9, no. 4, pp. 375–382, 1977.
- [3] V. Khatibi and G. A. Montazer, "Intuitionistic fuzzy set vs. fuzzy set application in medical pattern recognition," *Artificial Intelligence in Medicine*, vol. 47, no. 1, pp. 43–52, 2009.
- [4] S. Cateni, M. Vannucci, and V. Colla, "Industrial multiple criteria decision making problems handled by means of fuzzy inference-based decision support systems," in *Proceedings of the 4th International Conference on Intelligent Systems, Modelling and Simulation, ISMS 2013*, pp. 12–17, Thailand, January 2013.
- [5] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.
- [6] F. Herrera and J. L. Verdegay, "Linguistic assessments in group decision," in *Proceedings of the First European Congress on Fuzzy and Intelligent Technologies*, pp. 941–948, Aachen, Germany, 1993.
- [7] F. Herrera, E. Herrera-Viedma, and J. L. Verdegay, "A model of consensus in group decision making under linguistic assessments," *Fuzzy Sets and Systems*, vol. 78, no. 1, pp. 73–87, 1996.
- [8] F. Herrera, E. Herrera-Viedma, and J. L. Verdegay, "Direct approach processes in group decision making using linguistic OWA operators," *Fuzzy Sets and Systems*, vol. 79, no. 2, pp. 175–190, 1996.
- [9] Z. S. Xu, "An approach based on the uncertain LOWG and induced uncertain LOWG operators to group decision making with uncertain multiplicative linguistic preference relations," *Decision Support Systems*, vol. 41, no. 2, pp. 488–499, 2006.
- [10] Z. Xu, *Uncertain Multi-Attribute Decision Making: Methods and Applications*, Tsinghua University, 2015.
- [11] X.-F. Wang, J.-Q. Wang, and W.-E. Yang, "Multi-criteria group decision making method based on intuitionistic linguistic aggregation operators," *Journal of Intelligent & Fuzzy Systems. Applications in Engineering and Technology*, vol. 26, no. 1, pp. 115–125, 2014.
- [12] H. Guan, S. Guan, and A. Zhao, "Intuitionistic fuzzy linguistic soft sets and their application in multi-attribute decision-making," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 31, no. 6, pp. 2869–2879, 2016.
- [13] R. R. Yager, "Multi-criteria decision making with ordinal/linguistic intuitionistic fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 3, pp. 590–599, 2015.
- [14] Y. Pang and W. Yang, "Some hesitant intuitionistic fuzzy linguistic distance measures," *International Journal of Innovative Computing, Information and Control*, vol. 11, no. 5, pp. 1573–1586, 2015.
- [15] W. Yang, Y. Pang, J. Shi, and C. Wang, "Linguistic hesitant intuitionistic fuzzy decision-making method based on VIKOR," *Neural Computing and Applications*, pp. 1–14, 2016.
- [16] S. Santini and R. Jain, "Similarity measures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871–883, 1999.
- [17] S. M. Mulekar and C. S. Brown, "Distance and similarity measures," in *Encyclopedia of Social Network Analysis and Mining*, pp. 385–400, Springer, New York, NY, USA, 2015.
- [18] J. Ye and Q. Zhang, "Single valued neutrosophic similarity measures for multiple attribute decision-making," *Neutrosophic Sets and Systems*, 2015.
- [19] S.-H. Cheng, S.-M. Chen, and W.-S. Jian, "Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures," *Information Sciences*, vol. 327, pp. 272–287, 2016.
- [20] J. Ye, "Cosine similarity measures for intuitionistic fuzzy sets and their applications," *Mathematical and Computer Modelling*, vol. 53, no. 1-2, pp. 91–97, 2011.
- [21] L. Zhou, Z. Tao, H. Chen, and J. Liu, "Intuitionistic fuzzy ordered weighted cosine similarity measure," *Group Decision & Negotiation*, vol. 23, no. 4, pp. 879–900, 2014.
- [22] D. Liu, X. Chen, and D. Peng, "Interval-valued intuitionistic fuzzy ordered weighted cosine similarity measure and its application in investment decision-making," *Complexity*, vol. 2017, Article ID 1891923, 11 pages, 2017.
- [23] J.-Q. Wang, J.-T. Wu, J. Wang, H.-Y. Zhang, and X.-H. Chen, "Multi-criteria decision-making methods based on the Hausdorff distance of hesitant fuzzy linguistic numbers," *Soft Computing*, vol. 20, no. 4, pp. 1621–1633, 2016.

- [24] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.
- [25] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.
- [26] K. Atanassov and G. Gargov, "Interval valued intuitionistic fuzzy sets," *Fuzzy Sets and Systems*, vol. 31, no. 3, pp. 343–349, 1989.
- [27] K. T. Atanassov, *Intuitionistic Fuzzy Sets*, Physica-Verlag, New York, NY, USA, 1999.

Research Article

A Hybrid Approach for Modular Neural Network Design Using Intercriteria Analysis and Intuitionistic Fuzzy Logic

Sotir Sotirov ¹, Evdokia Sotirova,¹ Vassia Atanassova,² Krassimir Atanasov,^{1,2} Oscar Castillo ³, Patricia Melin,³ Todor Petkov,¹ and Stanimir Surchev¹

¹Intelligent Systems Laboratory, “Prof. Dr. Asen Zlatarov” University, Burgas, Bulgaria

²Bioinformatics and Mathematical Modelling Department, IBPhBME-Bulgarian Academy of Sciences, Sofia, Bulgaria

³Tijuana Institute of Technology, Tijuana, BC, Mexico

Correspondence should be addressed to Oscar Castillo; ocastillo@tectijuana.mx

Received 20 September 2017; Revised 30 December 2017; Accepted 8 March 2018; Published 19 April 2018

Academic Editor: Enzo Pasquale Scilingo

Copyright © 2018 Sotir Sotirov 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.

Intercriteria analysis (ICA) is a new method, which is based on the concepts of index matrices and intuitionistic fuzzy sets, aiming at detection of possible correlations between pairs of criteria, expressed as coefficients of the positive and negative consonance between each pair of criteria. Here, the proposed method is applied to study the behavior of one type of neural networks, the modular neural networks (MNN), that combine several simple neural models for simplifying a solution to a complex problem. They are a tool that can be used for object recognition and identification. Usually the inputs of the MNN can be fed with independent data. However, there are certain limits when we may use MNN, and the number of the neurons is one of the major parameters during the implementation of the MNN. On the other hand, a high number of neurons can slow down the learning process, which is not desired. In this paper, we propose a method for removing part of the inputs and, hence, the neurons, which in addition leads to a decrease of the error between the desired goal value and the real value obtained on the output of the MNN. In the research work reported here the authors have applied the ICA method to the data from real datasets with measurements of crude oil probes, glass, and iris plant. The method can also be used to assess the independence of data with good results.

1. Introduction

One of the open and important questions in biology is the ability of biological systems to adapt to new environments, a concept termed evolvability [1]. A typical feature of evolvability is the fact that many biological systems have modularity; especially many biological processes and structures can be modeled as networks, such as metabolic pathways, gene regulation, protein interactions, and brains [1–5]. This feature has motivated important concepts in intelligent systems, such as modular neural network and evolutionary computation.

Neural networks are considered modular if they are comprised of highly connected clusters of nodes that are connected to nodes in other clusters [4, 6, 7]. Despite importance and continuous research in this area, there is no agreement on why modular biological systems can evolve

[4, 8, 9]. There is evidence that modular systems look more adaptable in nature [10] than the monolithic networks [11, 12]. Consequently, there are many papers dedicated to this problem, for example, the work in [12].

In this paper, we introduce a hybrid combination between the intercriteria analysis (ICA, see [13–18]) method and modular neural network models. The ICA employs the apparatus of the intuitionistic fuzzy sets (IFS) for detecting possible correlations between pairs of criteria. Introduced in [19], IFSs are one of the extensions of Zadeh’s fuzzy sets [20]. In contrast to fuzzy sets, IFS [21–24] have two degrees: of membership (validity, etc., μ_A) and of nonmembership (nonvalidity, etc., ν_A), so that for each element x of the universe, over which an IFS A is defined, the following inequality is valid: $0 \leq \mu_A(x) + \nu_A(x) \leq 1$. In this case, a pair $\langle a, b \rangle$, where $a, b, a + b \in [0, 1]$, is called an intuitionistic fuzzy pair (IF pair). The ICA method produces the so-called positive and negative

consonance coefficients between the different criteria used for evaluation of different objects.

The main contribution of the paper is the proposed hybrid approach combining intuitionistic fuzzy logic (through the ICA method) with modular neural networks for designing a powerful neural model for classification. The neural network model is tested with benchmark problems and a real world case to show the advantages of the proposed approach. As regards existing works that could be considered similar to this one, we can mention that intuitionistic fuzzy logic has not been considered in conjunction with modular neural networks previously, so it can be considered an original contribution to the area of computational intelligence that combines the advantages of the two methods. For the purpose of testing the proposed method for preprocessing the information going into MNNs, we use data from the LUKOIL Neftochim Burgas AD from the measurements of a set of crude oil probes (objects, in terms of ICA) against a set of technological properties (criteria, in terms of ICA), which precedes and conditions the process of production of petrochemical products from the crude oil [25], dataset for iris plant [26], and glass types [27].

The remainder of the paper is organized as follows. In Section 2 some short remarks about the intercriteria analysis method are given, which is based on intuitionistic fuzzy logic. Section 3 describes basic concepts about modular neural networks. Section 4 describes the simulations and a discussion of the results. Finally, Section 5 offers the conclusions and outlines future work in this area.

2. Short Remarks on the Index Matrices and Intercriteria Analysis Method

As we mentioned above, the ICA method [13, 14] is based on two main concepts: intuitionistic fuzzy sets and index matrices. A brief description is offered below for completeness. Index matrices allow summarizing the criteria relevant to a particular decision making problem.

Let I be a fixed set of indices and let R be the set of the real numbers. An index matrix (IM) with sets of indices K and L ($K, L \subset I$) is defined by (see [13])

$$\left[K, L, \{a_{k_i, l_j}\} \right] \equiv \begin{array}{c|cccc} & l_1 & l_2 & \cdots & l_n \\ \hline k_1 & a_{k_1, l_1} & a_{k_1, l_2} & \cdots & a_{k_1, l_n} \\ k_2 & a_{k_2, l_1} & a_{k_2, l_2} & \cdots & a_{k_2, l_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_m & a_{k_m, l_1} & a_{k_m, l_2} & \cdots & a_{k_m, l_n} \end{array} \quad (1)$$

where $K = \{k_1, k_2, \dots, k_m\}$, $L = \{l_1, l_2, \dots, l_n\}$, for $1 \leq i \leq m$, and $1 \leq j \leq n : a_{k_i, l_j} \in R$.

For any two IMs, a series of relations, operations, and operators have been defined. The theory behind the IMs is described in a more detailed fashion in [13].

Here, following the description of the ICA approach, given by [14], we will start with the IM called M with index sets with m rows $\{O_1, \dots, O_m\}$ and n columns $\{C_1, \dots, C_n\}$, where for every p, q ($1 \leq p \leq m, 1 \leq q \leq n$),

O_p is an evaluated object, C_q is an evaluation criterion, and e_{O_p, C_q} is the evaluation of the p th object against the q th criterion, defined as a real number that is comparable according to relation R with all the remaining elements of the IM M .

$$M = \begin{array}{c|cccccc} & C_1 & \cdots & C_k & \cdots & C_l & \cdots & C_n \\ \hline O_1 & e_{O_1, C_1} & \cdots & e_{O_1, C_k} & \cdots & e_{O_1, C_l} & \cdots & e_{O_1, C_n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ O_i & e_{O_i, C_1} & \cdots & e_{O_i, C_k} & \cdots & e_{O_i, C_l} & \cdots & e_{O_i, C_n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ O_j & e_{O_j, C_1} & \cdots & e_{O_j, C_k} & \cdots & e_{O_j, C_l} & \cdots & e_{O_j, C_n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ O_m & e_{O_m, C_1} & \cdots & e_{O_m, C_j} & \cdots & e_{O_m, C_l} & \cdots & e_{O_m, C_n} \end{array} \quad (2)$$

From the requirement for comparability above, it follows that for each i, j, k the relation $R(e_{O_i, C_k}, e_{O_j, C_k})$ holds. The relation R has a dual relation \bar{R} , which is true in the cases when the relation R is false, and vice versa. For instance, if R is "greater," the dual relation \bar{R} is "less."

For the requirements of the proposed method, pairwise comparisons between every two different criteria are made along all evaluated objects. During the comparison, a counter is maintained for the number of times when the relation R holds, as well as another counter for the dual relation.

Let $S_{k,l}^{\mu}$ be the number of cases in which the relations $R(e_{O_i, C_k}, e_{O_j, C_k})$ and $R(e_{O_i, C_l}, e_{O_j, C_l})$ are simultaneously satisfied. Let also $S_{k,l}^{\nu}$ be the number of cases in which the relations $R(e_{O_i, C_k}, e_{O_j, C_k})$ and the dual $\bar{R} = (e_{O_i, C_l}, e_{O_j, C_l})$ are simultaneously satisfied. As the total number of pairwise comparisons between the objects is given by $m(m-1)/2$, it can be verified that the following inequalities hold:

$$0 \leq S_{k,l}^{\mu} + S_{k,l}^{\nu} \leq \frac{m(m-1)}{2}. \quad (3)$$

For every k, l , such that $1 \leq k \leq l \leq n$ and for $m \geq 2$ two numbers are defined:

$$\mu_{C_k, C_l} = 2 \frac{S_{k,l}^{\mu}}{m(m-1)}, \quad (4)$$

$$\nu_{C_k, C_l} = 2 \frac{S_{k,l}^{\nu}}{m(m-1)}.$$

The pair constructed from these two numbers plays the role of the intuitionistic fuzzy evaluation of the relations that can be established between any two criteria C_k and C_l . In this way, the IM M that relates evaluated objects with evaluating criteria can be transformed to another IM M^* that gives the relations detected among the criteria, where stronger

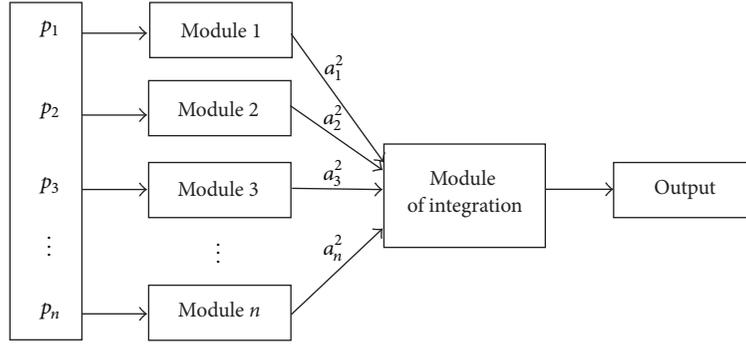


FIGURE 1: The structure of the MNN.

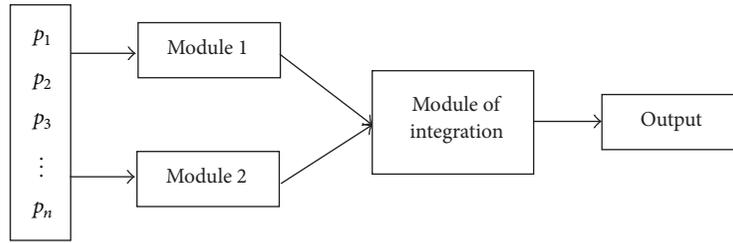


FIGURE 2: The structure of the MNN with 2 modules.

correlation exists where the first component μ_{C_k, C_l} is higher while the second component ν_{C_k, C_l} is lower.

$$M^* = \begin{array}{c|ccc} & C_1 & \cdots & C_n \\ \hline C_1 & \langle \mu_{C_1, C_1}, \nu_{C_1, C_1} \rangle & \cdots & \langle \mu_{C_1, C_n}, \nu_{C_1, C_n} \rangle \\ \cdots & \cdots & \cdots & \cdots \\ C_n & \langle \mu_{C_n, C_1}, \nu_{C_n, C_1} \rangle & \cdots & \langle \mu_{C_n, C_n}, \nu_{C_n, C_n} \rangle \end{array} \quad (5)$$

From practical considerations, it has been more flexible to work with two IMs M^μ and M^ν , rather than with the IM M^* of IF pairs. IM M^μ contains as elements the first components of the IFPs of M^* , while M^ν contains the second components of the IFPs of M^* . Once the intercriteria pairs have been calculated, for example, using the software described in [28], the question arises about defining the thresholds against which the membership and the nonmembership parts are evaluated [29, 30].

As has been discussed in some publications on ICA, for example, in [31, 32], the ICA results are very close to those obtained with the correlation analyses of Spearman, Pearson, and Kendall. It is worth noting the so far empirically observed fact that when in the data there are mistakes (e.g., shift of the decimal separator) these three correlation analyses give a larger deviation of the value than ICA; that is, ICA is less sensitive, so the use of them together can be used as a way of detecting errors in the input data.

3. Modular Neural Networks

Modular neural networks [33, 34] are one of the models that can be used for object recognition, classification, and

identification (see Figure 1). A modular neural network can be viewed as a set of monolithic neural networks [35–37] that deal with a part of a problem, and then their individual outputs are combined by an integration unit to form a global solution to the complete problem. The main idea is that a complex problem can be divided into simpler subproblems that can be solved by simpler neural networks and then the total solution will be a combination of the outputs of the simple monolithic neural networks.

In the proposed hybrid approach each of the MNN Modules takes as its input the result of the application of ICA method over three datasets (dataset for production of petrochemical products from the crude oil [25], iris plants [26], and glass types [27]). Every module is a two-layer Multilayer Perceptron and the output of the second layer of the ANN is a_i^2 (for $i \in \{1, \dots, n\}$), where n is the maximal number of modules.

The output O is calculated according to the following equation, which basically performs a weighted integration of the module outputs:

$$O = \frac{\sum_{i=1}^n a_i^2 g_i k_i}{\sum_{i=1}^n g_i}, \quad (6)$$

where a_i^2 is output of the module $i \in [1, 2, \dots, n]$; g_i is average deviation of the output values of the module i ; k_i is coefficient of existence of module i . The coefficient of existence of every module shows the presence/absence of the respective module, with respect to the need of that module in the particular case.

For illustration purposes and simplifying calculations, we can choose to reduce the structure of the MNN and use 2

TABLE 1: Membership parts of the intuitionistic fuzzy pairs, giving the intercriteria correlations between measurements of crude oil probes.

μ	1	2	3	4	5	6	7	8
1	1.000	0.699	0.770	0.658	0.956	0.176	0.446	0.703
2	0.699	1.000	0.787	0.597	0.676	0.408	0.640	0.775
3	0.770	0.787	1.000	0.777	0.728	0.394	0.665	0.921
4	0.658	0.597	0.777	1.000	0.627	0.468	0.674	0.771
5	0.956	0.676	0.728	0.627	1.000	0.134	0.404	0.661
6	0.176	0.408	0.394	0.468	0.134	1.000	0.730	0.473
7	0.446	0.640	0.665	0.674	0.404	0.730	1.000	0.743
8	0.703	0.775	0.921	0.771	0.661	0.473	0.743	1.000

TABLE 2: Nonmembership parts of the intuitionistic fuzzy pairs, giving the intercriteria correlations between measurements of crude oil probes.

ν	1	2	3	4	5	6	7	8
1	0.000	0.288	0.217	0.326	0.042	0.822	0.552	0.295
2	0.288	0.000	0.204	0.391	0.312	0.580	0.348	0.213
3	0.217	0.204	0.000	0.212	0.261	0.595	0.325	0.068
4	0.326	0.391	0.212	0.000	0.359	0.518	0.312	0.215
5	0.042	0.312	0.261	0.359	0.000	0.866	0.596	0.339
6	0.822	0.580	0.595	0.518	0.866	0.000	0.270	0.527
7	0.552	0.348	0.325	0.312	0.596	0.270	0.000	0.257
8	0.295	0.213	0.068	0.215	0.339	0.527	0.257	0.000

modules from the structure from Figure 2 ($k_1 = 1$, $k_2 = 1$, $k_3 = 0, \dots, k_n = 0$).

The first module considers all independent inputs (weak dissonance, dissonance, and strong dissonance). The second module takes the inputs that have strong negative consonance, negative consonance, and weak negative consonance, and weak positive consonance, positive consonance, and strong positive consonance.

In the second module, we can reduce some of the inputs if they have very strong positive consonance. In this case, we can remove one of the inputs. In the case that they have strong negative consonance, this means that we can also remove some of the inputs. Other configurations for the modular neural network are possible, depending on how the inputs are selected.

4. Discussion of Results

For verifying the accuracy of the proposed method the data of the real dataset with measurements of crude oil probes, glass, and iris plants were used.

4.1. Testing with Data for Crude Oil Probes. For the learning process of MNN, we set the following parameters: performance (MSE) = 0.00001; validation check = 15. The dataset was divided into three different parts: training (from 1 to 100); validation (from 101 to 120), and testing (from 121 to 140). For the purpose of learning algorithms, one of the variants of the backpropagation algorithm is used, namely, Levenberg-Marquardt. As a target for the model, the values from the databases were used.

In the MNN, the data for 140 crude oil probes, measured against 8 physical properties (“criteria” in the ICA terminology) were used (for the input data, see [34]). The eight criteria are as follows: (1) density at 15°C g/cm³; (2) 10% (v/v) ASTM D86 distillation, °C; (3) 50% (v/v) ASTM D86 distillation, °C; (4) 90% (v/v) ASTM D86 distillation, °C; (5) refractive index at 20°C; (6) H₂ content, % (m/m); (7) aniline point, °C; (8) molecular weight g/mol. Using the ICA approach, we are seeking for correlations between these eight criteria on the basis of the 140 crude oil probes (“objects” in ICA). Using the ICA software [28] applied to the whole 140 × 8 matrix with measurements, we obtain 28 IF pairs, giving the pairwise correlations between the eight criteria. For easier processing of the result of ICA application, the output is given in the form of two IMs, containing the membership and the nonmembership parts of the IF correlations discovered between each pair of criteria (Tables 1 and 2), which in Table 3 are sorted in descending order according to the membership part of the intercriteria pairs, from strong positive consonance (i.e., pair 1-5 with degree ⟨0.956; 0.042⟩), through dissonance, to negative consonance (i.e., pair 5-6 with degree ⟨0.134; 0.866⟩).

On this basis, we separate these 28 intercriteria pairs in two groups, where *Module 1* of the MNN takes as input the dissonant pairs, and *Module 2* takes as input the consonant pairs, either positive or negative. In this way, we aim to reduce the number of input parameters of the MNN, yet keeping high enough level of precision.

In Table 4 we are presenting the values from 22 neural network simulations and the identifiers of criteria that we provide at the input on each module. These 22 simulations

TABLE 3: Relations among the criteria with specific values.

IF pair	Criteria	Type of intercriteria relation
$\langle 0.956; 0.042 \rangle$	1-5	Strong positive consonance
$\langle 0.921; 0.068 \rangle$	3-8	Positive consonance
$\langle 0.787; 0.204 \rangle$	2-3	Weak positive consonance
$\langle 0.777; 0.212 \rangle$	3-4	Weak positive consonance
$\langle 0.771; 0.215 \rangle$	4-8	Weak positive consonance
$\langle 0.770; 0.217 \rangle$	1-3	Weak positive consonance
$\langle 0.743; 0.257 \rangle$	7-8	Weak dissonance
$\langle 0.730; 0.270 \rangle$	6-7	Weak dissonance
$\langle 0.728; 0.261 \rangle$	3-5	Weak dissonance
$\langle 0.703; 0.295 \rangle$	1-8	Weak dissonance
$\langle 0.699; 0.288 \rangle$	1-2	Weak dissonance
$\langle 0.676; 0.312 \rangle$	2-5	Weak dissonance
$\langle 0.674; 0.312 \rangle$	4-7	Weak dissonance
$\langle 0.665; 0.325 \rangle$	3-7	Dissonance
$\langle 0.661; 0.339 \rangle$	5-8	Dissonance
$\langle 0.658; 0.326 \rangle$	1-4	Dissonance
$\langle 0.640; 0.348 \rangle$	2-7	Dissonance
$\langle 0.627; 0.359 \rangle$	4-5	Dissonance
$\langle 0.597; 0.391 \rangle$	2-4	Dissonance
$\langle 0.473; 0.527 \rangle$	6-8	Strong dissonance
$\langle 0.468; 0.518 \rangle$	4-6	Strong dissonance
$\langle 0.446; 0.552 \rangle$	1-7	Strong dissonance
$\langle 0.408; 0.580 \rangle$	2-6	Dissonance
$\langle 0.404; 0.596 \rangle$	5-7	Dissonance
$\langle 0.394; 0.595 \rangle$	3-4	Dissonance
$\langle 0.394; 0.594 \rangle$	3-6	Dissonance
$\langle 0.176; 0.822 \rangle$	1-6	Weak negative consonance
$\langle 0.134; 0.866 \rangle$	5-6	Negative consonance

are all the possible combinations of criteria, where *Module 2* obtains as input at least one criterion, starting with all of the consonant criteria (1, 3, 5, 6, and 8 in row 1) down to at least one of the consonant criteria (1, 3, 5, 6, or 8, in rows 18–22). Based on the values from Table 3, we offer the use of the relations of positive consonance and negative consonance, since both affect in the same way the inputs of the neural network. The parameters g_1 , g_2 , and g_3 are the average deviations of the output values of *Module 1*, *Module 2*, and whole neural network, respectively. A detailed discussion of the results follows in Section 4.4.

4.2. Testing with Data for Iris Plants. As another test case for the proposed method, we also consider the dataset [26] that contains four parameters (criteria): (1) sepal length in cm, (2) sepal width in cm, (3) petal length in cm, and (4) petal width in cm. There are 3 classes of 50 instances each [26], where each class refers to a type of iris plant. The targets are *Iris setosa*, 1, *Iris versicolor*, 2, and *Iris virginica*, 3.

The ICA method was applied to the 150×4 matrix in the same fashion as previously (see Tables 5, 6, and 7), and the results are given in Table 5, sorted in descending order according to the membership part of the intercriteria pairs.

In the neural network inputs were assumed with the following parameters: parameter 2 (in the first neural network input) and parameters 1, 3, and 4 in consonance (in the second neural network input). In Table 8 the values from 7 neural network simulations are presented and a number of parameters that we defined for each module are described.

4.3. Testing with Data for Glass. Finally, as a third case for testing the proposed approach we consider the benchmark glass data. In this case, in the input of this modular neural network, we use the experimental data from [27] for obtaining the type of glass. We work with data for 214 building and vehicle window glasses, measured against 9 criteria: (1) RI: refractive index; (2) Na: sodium (unit measurement: weight percent in corresponding oxide); (3) Mg: magnesium; (4) Al: aluminium; (5) Si: silicon; (6) K: potassium; (7) Ca: calcium; (8) Ba: barium; (9) Fe: iron. In the output (as a target) we give information about the glass type.

As was done in the previous sections, the input parameters (criteria) are divided in two groups: parameters in dissonance (parameters 2, 3, 4, 5, 6, and 9) and parameters in consonance (see Table 9), with the results sorted in

TABLE 4: Results from the simulations.

Number	Description	Number of the inputs	g_1	g_2	g_3	Number of weight coefficients
1	Module 1—2, 4, and 7 Module 2—1, 3, 5, 6, and 8	8	2.2971	2.1991	2.2330	315
2	Module 1—2, 4, and 7 Module 2—3, 5, 6, and 8 (without 1)	7	2.2139	2.1650	2.1562	280
3	Module 1—2, 4, and 7 Module 2—1, 5, 6, and 8 (without 3)	7	2.2023	2.1498	2.1672	280
4	Module 1—2, 4, and 7 Module 2—1, 3, 6, and 8 (without 5)	7	2.2032	2.1559	2.1564	280
5	Module 1—2, 4, and 7 Module 2—1, 3, 5, and 8 (without 6)	7	2.1352	2.2455	2.1593	280
6	Module 1—2, 4, and 7 Module 2—1, 3, 5, and 6 (without 8)	7	2.1703	2.1578	2.1638	280
7	Module 1—2, 4, and 7 Module 2—5, 6, and 8 (without 1 and 3)	6	2.1844	2.1196	2.1457	245
8	Module 1—2, 4, and 7 Module 2—1, 3, and 6 (without 5 and 8)	6	2.1881	2.1500	2.1572	245
9	Module 1—2, 4, and 7 Module 2—1, 6, and 8 (without 3 and 5)	6	2.2057	2.2465	2.1572	245
10	Module 1—2, 4, and 7 Module 2—3, 5, and 6 (without 1 and 8)	6	2.1715	2.1531	2.1437	245
11	Module 1—2, 4, and 7 Module 2—3, 6, and 8 (without 1 and 5)	6	2.1679	2.1536	2.1370	245
12	Module 1—2, 4, and 7 Module 2—1, 3, and 5 (without 6 and 8)	6	2.2181	2.1415	2.1398	245
13	Module 1—2, 4, and 7 Module 2—1, 3, and 8 (without 5 and 6)	6	2.1883	2.1624	2.1551	245
14	Module 1—2, 4, and 7 Module 2—3 and 6 (without 1, 5, and 8)	5	2.1478	2.1885	2.1485	210
15	Module 1—2, 4, and 7 Module 2—3 and 5 (without 1, 6, and 8)	5	2.1542	2.1210	2.0914	210
16	Module 1—2, 4, and 7 Module 2—1 and 3 (without 5, 6, and 8)	5	2.1436	2.2551	2.1580	210
17	Module 1—2, 4, and 7 Module 2—3 and 8 (without 1, 5, and 6)	5	2.1777	2.2327	2.1981	210
18	Module 1—2, 4, and 7 Module 2—1 (without 3, 5, 6, and 8)	4	2.2006	7.0510	5.6472	210
19	Module 1—2, 4, and 7 Module 2—3 (without 1, 5, 6, and 8)	4	2.1554	6.6392	5.2857	175
20	Module 1—2, 4, and 7 Module 2—5 (without 1, 3, 6, and 8)	4	2.1856	6.8357	5.4230	175
21	Module 1—2, 4, and 7 Module 2—6 (without 1, 3, 5, and 8)	4	2.1873	3.6355	2.7732	175
22	Module 1—2, 4, and 7 Module 2—8 (without 1, 3, 5, and 8)	4	2.2109	6.1718	4.9165	175

descending order according to the membership part of the intercriteria pairs.

In Table 10, the values from 7 neural network simulations are presented with the respective identifiers of the criteria that are feeding the inputs of each module.

The information is divided into two groups: the first part is the group of independent evaluation criteria (see Tables 11 and 12): criteria 2, 4, and 7 (see Table 4 for the crude oil probes

data), criterion 2 (see Table 5 for the iris plant data), and criteria 2, 3, 4, 5, 6, and 9 (see Table 8 for the glass types data). These data were used on the inputs of the first module of the neural networks. The other criteria (dependent parameters) were considered on the inputs of the second module of the neural networks.

The inputs on the first module are not removable, because the ICA approach tests the independence of the criteria. The

TABLE 5: Relations among the criteria.

IF pair	Criteria	Type of intercriteria relation
(0.843; 0.538)	3-4	Weak positive consonance
(0.819; 0.504)	1-3	Weak positive consonance
(0.764; 0.486)	1-4	Weak positive consonance
(0.416; 0.139)	1-2	Dissonance
(0.366; 0.121)	2-4	Dissonance
(0.364; 0.070)	2-3	Dissonance

TABLE 6: Membership parts of the intuitionistic fuzzy pairs, giving the intercriteria correlations between measurements of iris plant.

μ	1	2	3	4
1	1.000	0.416	0.819	0.764
2	0.416	1.000	0.364	0.366
3	0.819	0.364	1.000	0.843
4	0.764	0.366	0.843	1.000

TABLE 7: Nonmembership parts of the intuitionistic fuzzy pairs, giving the intercriteria correlations between measurements of iris plant.

ν	1	2	3	4
1	0.000	0.486	0.121	0.139
2	0.486	0.000	0.538	0.504
3	0.121	0.538	0.000	0.070
4	0.139	0.504	0.070	0.000

results from applying the ICA method over tree types of data show that most of the parameters are selected in a proper way (criteria in dissonance). For some of the parameters that were provided on the inputs of the second module, an approach for reasonable elimination of some of these criteria can be adopted.

4.4. Discussion. In the first step we use neural networks without any removal of information and without removing of the inputs. The corresponding results can be found on the first rows in Tables 4, 8, and 10.

On the second step the data that have the highest and lowest consonance coefficients (strong positive consonance, positive consonance, strong negative consonance, and negative consonance) were removed. For example, in Table 4 we use parameters 2, 4, and 7 in *Module 1* and parameters 3, 5, 6, and 8 (without 1) in *Module 2*. In the study, parameter 1 is removed due to the high value of membership coefficient $\mu = 0.956$ (between parameters 1 and 5). After simulating the neural network an average deviation of the MNN $g_3 = 2.1562$ ($g_1 = 2.2139$, $g_2 = 2.1650$) is obtained.

The removed parameters do not have substantial influence on the result, because of decreasing the number of the weight coefficients, and along with this the error of the output values also decreases.

In the same way the second group of parameters (from the pairs) were successively removed. This process continues until finishing the pairs in the highest and lowest consonance.

The next step is to remove two parameters, for example, in row 5 (Table 8) the situation with all parameters on *Module 1*, and parameter 4 (without 1 and 3) in *Module 2* was presented. In the study, parameters 1 and 3 are removed due to the high value of membership coefficient $\mu = 0.81852$ (between parameters 1 and 3). After simulation of the neural network, an average deviation of the MNN $g_3 = 0.5846$ ($g_1 = 0.5725$, $g_2 = 0.0631$) is obtained. The removed parameters 1 and 3 do not have substantial influence on the result, because of decreasing the number of the weight coefficients, and along with this the error of the output values also decreases.

The process can continue iteratively in the same way. In summary, a new approach for building modular neural networks based on ICA, which uses intuitionistic fuzzy logic, has been proposed and tested with three benchmark databases. The simulation results are good, in this way verifying the advantages of the hybrid method.

In Table 13 comparative analysis of the results of the three different datasets is made. The table includes a short description of each dataset, the initial number of inputs (number of the inputs before reduction), the number of inputs after reduction, and average deviation of the output values of the neural networks (g_3), as well as minimal value of average deviation of the output values of the neural networks (g_3 , min), the number of weight coefficients without reduction and with reduction, the percentage ratio between the average deviation of the output values of the neural networks and the minimal value of average deviation of the output values of the neural networks, respectively, and the number of weight coefficients without and with reduction.

From the data in the table, it is seen that reducing the number of input parameters and, respectively, the number of inputs of the MNN, the value of average deviation of the output values of the MNN decreases (93,65%–98,2%). On this basis, we further observe decrease of the number of weight coefficients (60%–80%).

5. Conclusions

The number of the neurons is one of the major parameters that need to be defined during the realization of the MNN. Of course, the higher number of neurons in a neural network has the effect of slowing down the learning process. Here, we use the intuitionistic fuzzy logic based method of intercriteria analysis to reduce the number of input parameters of the modular neural network. This leads to reduction of the weight matrices and thus allows implementation of the neural network in limited hardware, saving time and resources in training. The method can also be used to assess the independence of the criteria against which data are measured. Three different real datasets with measurements of glass types, crude oil probes, and iris plant specimens were used to verify the accuracy and efficiency of the proposed approach. As future work, we envision using the proposed approach with other types of complex problems or more real life situations, like in the design of intelligent controllers for robotic systems

TABLE 8: Results from the simulations.

Number	Description	Number of the inputs	g_1	g_2	g_3	Number of weight coefficients
1	Module 1—2 Module 2—1, 3, 4	4	0.5725	0.0084	0.5726	175
2	Module 1—2 Module 2—3, 4 (without 1)	3	0.5725	0.0321	0.5749	140
3	Module 1—2 Module 2—1, 4 (without 3)	3	0.5725	0.6667	1.2898	140
4	Module 1—2 Module 2—1, 3 (without 4)	3	0.5725	0.0527	0.5793	140
5	Module 1—2 Module 2—4 (without 1, 3)	2	0.5725	0.0631	0.5846	105
6	Module 1—2 Module 2—3 (without 1, 4)	2	0.5725	0.0807	0.5907	105
7	Module 1—2 Module 2—1 (without 3, 4)	2	0.5725	0.6667	1.2898	105

TABLE 9: Relations among the criteria.

IF pair	Criteria	Type of intercriteria relation
$\langle 0.760; 0.234 \rangle$	1-7	Weak positive consonance
$\langle 0.564; 0.423 \rangle$	4-5	Dissonance
$\langle 0.547; 0.409 \rangle$	4-6	Strong dissonance
$\langle 0.532; 0.455 \rangle$	2-4	Strong dissonance
$\langle 0.532; 0.404 \rangle$	3-6	Strong dissonance
$\langle 0.527; 0.425 \rangle$	1-3	Strong dissonance
$\langle 0.513; 0.481 \rangle$	1-2	Strong dissonance
$\langle 0.497; 0.493 \rangle$	2-7	Strong dissonance
$\langle 0.488; 0.470 \rangle$	5-6	Strong dissonance
$\langle 0.441; 0.509 \rangle$	2-3	Strong dissonance
$\langle 0.419; 0.571 \rangle$	5-7	Dissonance
$\langle 0.410; 0.090 \rangle$	8-9	Dissonance
$\langle 0.384; 0.604 \rangle$	4-7	Dissonance
$\langle 0.383; 0.607 \rangle$	2-6	Dissonance
$\langle 0.371; 0.579 \rangle$	3-7	Dissonance
$\langle 0.368; 0.594 \rangle$	1-6	Dissonance
$\langle 0.353; 0.596 \rangle$	3-5	Dissonance
$\langle 0.318; 0.672 \rangle$	1-4	Weak dissonance
$\langle 0.316; 0.237 \rangle$	6-9	Weak dissonance
$\langle 0.314; 0.645 \rangle$	6-7	Weak dissonance
$\langle 0.311; 0.236 \rangle$	3-9	Weak dissonance
$\langle 0.304; 0.240 \rangle$	7-9	Weak dissonance
$\langle 0.300; 0.693 \rangle$	1-5	Weak dissonance
$\langle 0.299; 0.245 \rangle$	1-9	Weak dissonance
$\langle 0.288; 0.659 \rangle$	3-4	Weak dissonance
$\langle 0.272; 0.057 \rangle$	4-8	Weak dissonance
$\langle 0.262; 0.697 \rangle$	2-6	Weak dissonance
$\langle 0.257; 0.070 \rangle$	2-8	Weak dissonance
$\langle 0.253; 0.291 \rangle$	4-9	Weak dissonance
$\langle 0.253; 0.291 \rangle$	5-9	Weak dissonance
$\langle 0.210; 0.334 \rangle$	2-9	Weak negative consonance
$\langle 0.203; 0.123 \rangle$	5-8	Weak negative consonance
$\langle 0.162; 0.164 \rangle$	7-8	Weak negative consonance
$\langle 0.123; 0.202 \rangle$	1-8	Negative consonance
$\langle 0.113; 0.213 \rangle$	6-8	Negative consonance
$\langle 0.059; 0.249 \rangle$	3-8	Negative consonance

TABLE 10: Results from the simulations.

N:	Description	Number of the inputs	g_1	g_2	g_3	Number of weight coefficients
1	Module 1—2, 3, 4, 5, 6, 9 Module 2—1, 7, 8	9	0.7658	0.8796	1.6721	350
2	Module 1—2, 3, 4, 5, 6, 9 Module 2—7, 8 (without 1)	8	0.7658	0.8696	1.6584	315
3	Module 1—2, 3, 4, 5, 6, 9 Module 2—1, 8 (without 7)	8	0.7658	0.9277	1.7492	315
4	Module 1—2, 3, 4, 5, 6, 9 Module 2—1, 7 (without 8)	8	0.7658	1.5826	2.8488	315
5	Module 1—2, 3, 4, 5, 6, 9 Module 2—8 (without 1, 7)	7	0.7658	0.8756	1.6626	280
6	Module 1—2, 3, 4, 5, 6, 9 Module 2—7 (without 1, 8)	7	0.7658	1.7190	3.0898	280
7	Module 1—2, 3, 4, 5, 6, 9 Module 2—1 (without 7, 8)	7	0.7658	1.6378	2.9400	280

TABLE 11: Membership parts of the intuitionistic fuzzy pairs, giving the intercriteria correlations of glass.

μ	1	2	3	4	5	6	7	8	9
1	1.000	0.513	0.527	0.318	0.300	0.368	0.760	0.123	0.299
2	0.513	1.000	0.441	0.532	0.383	0.262	0.497	0.257	0.210
3	0.527	0.441	1.000	0.288	0.353	0.532	0.371	0.059	0.311
4	0.318	0.532	0.288	1.000	0.564	0.547	0.384	0.272	0.253
5	0.300	0.383	0.353	0.564	1.000	0.488	0.419	0.203	0.253
6	0.368	0.262	0.532	0.547	0.488	1.000	0.314	0.113	0.316
7	0.760	0.497	0.371	0.384	0.419	0.314	1.000	0.162	0.304
8	0.123	0.257	0.059	0.272	0.203	0.113	0.162	1.000	0.410
9	0.299	0.210	0.311	0.253	0.253	0.316	0.304	0.410	1.000

TABLE 12: Nonmembership parts of the intuitionistic fuzzy pairs, giving the intercriteria correlations.

ν	1	2	3	4	5	6	7	8	9
1	0.000	0.481	0.425	0.672	0.693	0.594	0.234	0.202	0.245
2	0.481	0.000	0.509	0.455	0.607	0.697	0.493	0.070	0.334
3	0.425	0.509	0.000	0.659	0.596	0.404	0.579	0.249	0.236
4	0.672	0.455	0.659	0.000	0.423	0.409	0.604	0.057	0.291
5	0.693	0.607	0.596	0.423	0.000	0.470	0.571	0.123	0.291
6	0.594	0.697	0.404	0.409	0.470	0.000	0.645	0.213	0.237
7	0.234	0.493	0.579	0.604	0.571	0.645	0.000	0.164	0.240
8	0.202	0.070	0.249	0.057	0.123	0.213	0.164	0.000	0.090
9	0.245	0.334	0.236	0.291	0.291	0.237	0.240	0.090	0.000

TABLE 13: Results from the simulations.

Number	Description	Number of the inputs before reduction	Number of the inputs after reduction	g_3	g_3 min	Number of weight coefficients without reduction	Number of weight coefficients with reduction	g_3 %	Number of weight coefficients %
1	Data with parameters of crude oil	8	5	2.2330	2.0914	315	210	93,65	66,67
2	Data of iris plant parameters	4	2	0.5726	0.5446	175	105	95,11	60
3	Data of glass parameters	9	7	1.6721	1.6626	350	280	98,2	80

or in the design of pattern recognition systems for human identification based on biometric measures. We believe that for real world situations some pertinent modifications or improvements to our model could be needed, but the essence of the solution to the problems is already provided in the proposed approach. As another leg of future investigation in the proposed direction, we consider a three-module MNN, which will be fed with the following inputs: one with the intercriteria pairs exhibiting positive consonance, another with the intercriteria pairs exhibiting negative consonance, and the third one with the pairs exhibiting dissonance.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors are grateful for the support provided by the project “New Instruments for Knowledge Discovery from Data, and Their Modelling,” funded by the National Science Fund, Bulgarian Ministry of Education, Youth and Science (no. DN-02-10/2016).

References

- [1] M. Pigliucci, “Is evolvability evolvable?” *Nature Reviews Genetics*, vol. 9, no. 1, pp. 75–82, 2008.
- [2] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*, Chapman & Hall/CRC, 2007.
- [3] S. B. Carroll, “Chance and necessity: The evolution of morphological complexity and diversity,” *Nature*, vol. 409, no. 6823, pp. 1102–1109, 2001.
- [4] G. P. Wagner, M. Pavlicev, and J. M. Cheverud, “The road to modularity,” *Nature Reviews Genetics*, vol. 8, no. 12, pp. 921–931, 2007.
- [5] R. Guimerà and L. A. N. Amaral, “Functional cartography of complex metabolic networks,” *Nature*, vol. 433, no. 7028, pp. 895–900, 2005.
- [6] H. Lipson, “Principles of modularity, regularity, and hierarchy for scalable systems,” *Journal of Biological Physics and Chemistry*, vol. 7, no. 4, pp. 125–128, 2007.
- [7] G. Striedter, *Principles of brain evolution*, Sinauer Associates, Sunderland, MA, 2005.
- [8] G. Wagner, J. Mezey, and R. Calabretta, *Modularity: understanding the development and evolution of complex natural systems*, Natural selection and the origin of Modules. Cambridge, MA: MIT Press, 2001.
- [9] C. Espinosa-Soto and A. Wagner, “Specialization can drive the evolution of modularity,” *PLoS Computational Biology*, vol. 6, no. 3, e1000719, 10 pages, 2010.
- [10] N. P. Suh, “The principles of design,” in *Spontaneous evolution of modularity and network motifs*, N. Kashtan and U. Alon, Eds., Proceedings of the National Academy of Sciences of the United States of America, 2005.
- [11] N. Kashtan, E. Noor, and U. Alon, “Varying environments can speed up evolution,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 34, pp. 13711–13716, 2007.
- [12] M. Parter, N. Kashtan, and U. Alon, “Environmental variability and modularity of bacterial metabolic networks,” *BMC Evolutionary Biology*, vol. 7, article 169, 2007.
- [13] K. T. Atanassov, *Index matrices: towards an augmented matrix calculus*, vol. 573 of *Studies in Computational Intelligence*, Springer, Cham, 2014.
- [14] K. Atanassov, D. Mavrov, and V. Atanassova, “InterCriteria decision making. A new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets,” *Issues in Intuitionistic Fuzzy Sets and Generalized Nets*, vol. 11, pp. 1–8, 2014.
- [15] “InterCriteria Research Portal,” <http://intercriteria.net/>.
- [16] S. Sotirov, V. Atanassova, E. Sotirova, V. Bureva, and D. Mavrov, “Application of the Intuitionistic Fuzzy InterCriteria Analysis Method to a Neural Network Preprocessing Procedure,” in *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15)*, Gijón, Spain., June 2015.
- [17] S. Todinova, D. Mavrov, and S. Krumova, “Blood plasma thermograms dataset analysis by means of intercriteria and correlation analyses for the case of colorectal cancer,” *International Journal Bioautomation*, vol. 20, no. 1, pp. 115–124, 2016.
- [18] O. Roeva, S. Fidanova, and M. Paprzycki, “InterCriteria analysis of ACO and GA hybrid algorithms,” in *Recent advances in computational optimization*, vol. 610 of *Stud. Comput. Intell.*, pp. 107–126, Springer, Cham, 2016.
- [19] K. Atanassov, “Intuitionistic Fuzzy Sets. VII ITKR Session, Sofia, 20-23 June 1983,” *International Journal Bioautomation*, vol. 20, no. S1, pp. S1–S6, 1983.
- [20] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [21] K. T. Atanassov, “Intuitionistic fuzzy sets,” *Fuzzy Sets and Systems*, vol. 20, no. 1, pp. 87–96, 1986.
- [22] K. T. Atanassov, *Intuitionistic Fuzzy Sets*, Physica-Verlag, New York, NY, USA, 1999.
- [23] K. T. Atanassov, “Intuitionistic Fuzzy Relations (IFRs),” in *On Intuitionistic Fuzzy Sets Theory*, vol. 283 of *Studies in Fuzziness and Soft Computing*, pp. 147–193, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [24] K. Atanassov, “Review and New Results on Intuitionistic Fuzzy Sets, Mathematical Foundations of Artificial Intelligence Seminar, Sofia, 1988,” *International Journal Bioautomation*, vol. 20, no. S1, pp. S7–S16, 2016.
- [25] D. Stratiev, I. Marinov, R. Dinkov et al., “Opportunity to improve diesel-fuel cetane-number prediction from easily available physical properties and application of the least-squares method and artificial neural networks,” *ENERGY & FUELS*, vol. 29, no. 3, pp. 1520–1533, 2015.
- [26] R. A. Fisher and M. Marshall, “Iris data set, UC Irvine Machine Learning Repository,” <https://archive.ics.uci.edu/ml/datasets/Iris>.
- [27] B. German and V. Spiehler, “Glass Identification Database, UC Irvine Machine Learning Repository,” <https://archive.ics.uci.edu/ml/datasets/Glass+Identification>.
- [28] D. Mavrov, “Software for InterCriteria Analysis: Implementation of the main algorithm,” *Notes on Intuitionistic Fuzzy Sets*, vol. 21, no. 2, pp. 77–86, 2015.
- [29] V. Atanassova, L. Doukovska, D. Mavrov, and K. Atanassov, “InterCriteria Decision Making Approach to EU Member States Competitiveness Analysis: Temporal and Threshold Analysis,” in *Intelligent Systems'2014*, vol. 322 of *Advances in Intelligent*

- Systems and Computing*, pp. 95–106, Springer International Publishing, Cham, 2015.
- [30] K. Atanassov, V. Atanassova, and G. Gluhchev, *InterCriteria Analysis: Ideas and problems, Notes on Intuitionistic Fuzzy Sets*, 2015.
- [31] S. Krumova, S. Todinova, and D. Mavrov, “Intercriteria analysis of calorimetric data of blood serum proteome,” *Biochimica et Biophysica Acta (BBA) - General Subjects*, vol. 1861, no. 2, pp. 409–417, 2017.
- [32] P. G. Marinov and S. Fidanova, (2015-2016) *Intercriteria and Correlation Analyses: Similarities, Differences and Simultaneous Use. Annual of “Informatics” Section, Union of Scientists in Bulgaria*.
- [33] P. Melin, *Modular Neural Networks and Type-2 Fuzzy Systems for Pattern Recognition*, vol. 389, Springer, Berlin, Heidelberg, 2012.
- [34] S. Sotirov, E. Sotirova, P. Melin, O. Castilo, and K. Atanassov, “Modular neural network preprocessing procedure with intuitionistic fuzzy InterCriteria analysis method,” *Advances in Intelligent Systems and Computing*, vol. 400, pp. 175–186, 2016.
- [35] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing, Boston, MA, 1996.
- [36] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan, New York, NY, USA, 1994.
- [37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

Research Article

Legendre Cooperative PSO Strategies for Trajectory Optimization

Lei Liu ¹, Yongji Wang ¹, Fuqiang Xie,² and Jiashi Gao¹

¹National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

²School of Electric Engineering, University of South China, Hengyang, Hunan, China

Correspondence should be addressed to Yongji Wang; wangyjch@hust.edu.cn

Received 29 December 2017; Revised 14 February 2018; Accepted 19 February 2018; Published 12 April 2018

Academic Editor: Jesus Medina

Copyright © 2018 Lei 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.

Particle swarm optimization (PSO) is a population-based stochastic optimization technique in a smooth search space. However, in a category of trajectory optimization problem with arbitrary final time and multiple control variables, the smoothness of variables cannot be satisfied since the linear interpolation is widely used. In the paper, a novel Legendre cooperative PSO (LCPSO) is proposed by introducing Legendre orthogonal polynomials instead of the linear interpolation. An additional control variable is introduced to transcribe the original optimal problem with arbitrary final time to the fixed one. Then, a practical fast one-dimensional interval search algorithm is designed to optimize the additional control variable. Furthermore, to improve the convergence and prevent explosion of the LCPSO, a theorem on how to determine the boundaries of the coefficient of polynomials is given and proven. Finally, in the numeral simulations, compared with the ordinary PSO and other typical intelligent optimization algorithms GA and DE, the proposed LCPSO has traits of lower dimension, faster speed of convergence, and higher accuracy, while providing smoother control variables.

1. Introduction

Swarm intelligence is a collective dynamic behavior of distributed, self-organized systems, natural or artificial, employed in work on artificial intelligence. It introduces many simple agents with very general rules to achieve an “intelligent” global optimal behavior. Swarm intelligence-based techniques can be used in a number of applications on optimization. The US military is investigating the swarming techniques to control unmanned vehicles. The European Space Agency is thinking about an orbital swarm for self-assembly and interferometry. NASA is investigating the use of swarm technology for planetary mapping.

In particular, trajectory optimization problem is one of the most important tasks in the preliminary design of the next generation of high speed vehicles, such as NASA's X-43 unmanned hypersonic vehicle (HV), and has a great effect on the choice of conceptual design [1]. It is a daunting work to get the solution of nonlinear optimal control problem with the arbitrary final time and multiconstraints.

Trajectory optimization with multiconstraints has been linked with some stochastic search algorithms. The typical approaches include genetic algorithms (GA) [2], differential evolution (DE) [3], and particle swarm optimization (PSO) [4–6]. Besides, several novel bionic optimization algorithms spring up in these years and show remarkable efficiency in the industrial domain, such as the honeybee mating optimization [7], harmony search algorithm [8], and ants swarm optimization [9]; however, they are rarely used in the aerospace field because of their excessive novelty. The PSO algorithm, as an optimization algorithm based on swarm intelligence, similar to GA and DE for their origination from population-based heuristic search mechanisms, has recently become more popular due to its simplicity and effectiveness. It is widely used because of its simple principle, small number of parameters to be adjusted, and easy realization. Considering that the dynamic parameter optimization needs to be transformed into the static format, it is validated to evaluate the trajectory parameters optimization using a spline function [10].

Under the guidance of thought of the hybrid algorithm [11], a novel method of trajectory optimization is proposed to improve the global convergence ability in this paper, the Legendre cooperative PSO (LCPSO) method, which is a kind of cooperative PSO based on Legendre orthogonal polynomials.

In LCPSO, the following improvements make it promising in solving complex problems. Firstly, the search space is divided into certain subspaces and different swarms are arranged to optimize the different parts of the search space. In this way, the optimized scale for each swarm can be reduced directly. The algorithm is suitable for the problems with larger scale and higher dimension.

Secondly, for the optimal control problem by PSO, the discretization is necessary before solving the parameter optimization problem. Herein, the Legendre orthogonal polynomial approximation can achieve higher smoothness of control variables with lower dimensions. Additionally, a theorem on how to find the precise range of optimal parameters is given and proven, as well as how to find the boundaries of the coefficient of polynomials. Therefore, the proposed LCPSO is expected to realize the optimization with higher accuracy and efficiency.

Thirdly, in order to solve the optimal control problems with arbitrary final time rather than the fixed one, the proposed Legendre orthogonal polynomial approximation method introduces an additional control variable to transcribe the original optimal problem to the one with fixed final time. Then, a traditional one-dimensional search method based on the interval analysis is proposed to optimize the additional control variable. This way, the specific optimal problem with single boundary can be solved.

Finally, we use two typical trajectory optimization problems to illustrate efficiency of the proposed LCPSO algorithm. One is the ascent trajectory optimization of X-43 hypersonic vehicle, and the other one is the classic optimal orbit transfer problem. The simulation results demonstrate the advantages of the LCPSO in terms of solution accuracy and convergence rate by comparing with some traditional intelligent optimization algorithms.

The organization of the remainder of this paper is as follows. Section 2 formulates the optimal control problem by 3-DOF mass point dynamics. Section 3 describes the algorithm of novel Legendre cooperative PSO and some properties. Section 4 presents the numerical simulations on the performance of the proposed optimal algorithms. Section 5 draws conclusions to the paper.

2. Problem Description

HV is one of the main workhorses for most of the nations of the world for scientific studies and military and commercial applications [1]. Efforts are ongoing in the 21st century to enhance flexibility and reliability and reduce the overall cost of such systems. Here, without loss of generality, we provide the proposed algorithm to solve a trajectory optimal

problem of HV. Its 3-DOF dynamics over a spherical earth are described by the following motion equations [12, 13]:

$$\begin{aligned}\dot{R} &= V \sin \gamma, \\ \dot{\theta} &= \frac{V \cos \gamma \sin \psi}{R \cos \phi}, \\ \dot{\phi} &= \frac{V \cos \gamma \cos \psi}{R}, \\ \dot{V} &= \frac{T \cos \alpha}{m} - D - \frac{\sin \gamma}{R^2}, \\ \dot{\gamma} &= \frac{1}{V} \left[\frac{T \sin \alpha}{m} + L + \left(V^2 - \frac{1}{R} \right) \frac{\cos \gamma}{R} \right], \\ \dot{m} &= f(Ma, \alpha, \delta).\end{aligned}\tag{1}$$

The 3-DOF dynamics described in (1) are dimensionless equations of motion with six state variables and two control variables. The real variables are normalized as follows:

$$\begin{aligned}R &= \frac{r}{r_0}, \\ V &= \frac{v}{\sqrt{g_0 r_0}}, \\ g &= \frac{g_0 r_0^2}{r^2},\end{aligned}\tag{2}$$

where r is the radial distance from the earth's center to the vehicle, θ is the longitude, ϕ is the latitude, v is the velocity of the vehicle, γ is the flight path angle, ψ is the velocity azimuth angle measured clockwise from the north, the control variables are the angle of attack α and the fuel throttle opening δ , m is the mass of the vehicle, and the terms D and L are the aerodynamic drag and lift acceleration, which are defined by

$$\begin{aligned}D &= \frac{1}{2} \frac{\rho v^2 C_D S_c}{mg_0}, \\ L &= \frac{1}{2} \frac{\rho v^2 C_L S_c}{mg_0}.\end{aligned}\tag{3}$$

Here, S_c is the reference area of the vehicle. The terms of $C_D(\alpha, Ma, \delta)$ and $C_L(\alpha, Ma, \delta)$ are the coefficients of drag and lift, which are also functions of α and δ .

The control variables α and δ are approximated by Legendre orthogonal polynomial functions, respectively. Assuming $L_N(x)$ is the Legendre polynomial of degree N , the continuous function $u(x)$ of variable x can be approximated as

$$u(x) = \sum_{n=0}^N b_n L_n(x).\tag{4}$$

This way, the dynamic parameter optimization problem is transformed into a static one before the PSO algorithm performed.

The evaluation of the performance index starts when the scramjet is launched. The vehicle releases from the boost

phase and then enters into the later ascent and cruise phase. Here, we define the switching time as t_k . By analyzing the dynamic characteristics of HV, the fuel throttle opening δ shall be minimized during the later ascent stage.

In this problem, the optimal time of ascent stage is free but satisfying some constraints. Thus, an additional control variable is introduced to transcribe the proposed problem to a fixed final time problem. The switching step is available and then the ascent trajectory can be divided into two phases. In the first phase, only the fuel throttle opening needs to be optimized to guarantee that it is a minimized constant value at the beginning of the second phase. In order to simplify the equations and problem, we give a reasonable assumption that the switching step satisfies $t_k = kt'_f$, $k \in [0, 1]$, with a fixed time interval $[t_0, t'_f]$, and the system's continuous state does not jump at each switching point. Therefore, the arbitrary terminal time optimal control problem is turned into a two-point boundary value problem with the fixed terminal time. The optimal time t_f would be obtained as long as all the terminal conditions are fulfilled.

The design space \mathbf{S} of HV ascent trajectory optimization using LCPSO can be defined by the following vector:

$$\mathbf{S} = (b_1^\alpha, \dots, b_N^\alpha, |b_1^\delta, \dots, b_N^\delta|, k). \quad (5)$$

Here, the left two parts in the vector \mathbf{S} represent the two groups of coefficients of the orthogonal polynomials which approximate the control variables α and δ , while the last parameter k represents the ratio between the switching step and the supposed final time. Hence, the problem space can be divided into three subspaces b^α , b^δ , and k . Then, in LCPSO, we provide two groups of particles to optimize the subspaces b^α and b^δ , respectively, and use a one-dimensional search method based on interval analysis to optimize the variable k .

To seek the solutions of both the continuous input $u(t)$ and the switching step t_k according to the given initial state $x(t_0) = x_0$, we define a performance function as follows:

$$J = \Phi(x(t_f)) + \int_{t_0}^{t_k} L(x(t), u(t)) dt + \int_{t_k}^{t_f} L'(x(t), u(t)) dt. \quad (6)$$

In this case, the above nonlinear optimal problem can be described as a basic optimal control problem of nonlinear Bolza type.

Considering that the performance function here is selected to minimize the fuel expendable ratio (FER) through the control variables α and δ , normally, we use the following relationship to describe it directly:

$$J = - \int_{t_0}^{t_f} \dot{m} dt = m_0 - m_f. \quad (7)$$

Furthermore, the ascent trajectory terminate conditions can be represented by the following inequalities:

$$r_f - \Delta r_{\text{down}} \leq r(t_f),$$

$$v_f - \Delta v_{\text{down}} \leq v(t_f),$$

$$\gamma_f - \Delta \gamma_{\text{down}} \leq \gamma(t_f) \leq \gamma_f + \Delta \gamma_{\text{up}}, \quad (8)$$

where $r(t_f)$, $v(t_f)$, and $\gamma(t_f)$ are three state variables of the terminal point, while the tolerance is given by the upper bound $\Delta \gamma_{\text{up}}$ and the lower bounds Δr_{down} , Δv_{down} , and $\Delta \gamma_{\text{down}}$.

The terminal conditions are added to the performance function to ensure that all the constraints are satisfied. Then, the performance function of LCPSO is defined as follows:

$$J(\alpha, \delta) = J + k_1 e^{(r_f - r(t_f))} + k_2 e^{(v_f - v(t_f))} + k_3 \gamma^2(t_f). \quad (9)$$

Here, in (9), the first part J represents the optimal index of minimum FER described in (7). The three terms $k_1 e^{(r_f - r(t_f))}$, $k_2 e^{(v_f - v(t_f))}$, and $k_3 \gamma^2(t_f)$ represent the terminal constraints index. $r(t_f)$ is the radial distance from the center of the earth to the vehicle and $v(t_f)$ represents the velocity at the final time, while the flight path angle index is $\gamma(t_f)$.

Considering that the proposed LCPSO provides two groups of particles based on Legendre orthogonal polynomial approximation, the assumption of the fixed time interval $[t_0, t'_f]$ can be transformed into a closed interval $[-1, 1]$ of Legendre orthogonal polynomials to facilitate subsequent processing. Meanwhile, the additional switching step t_k can make the optimal interval of δ more accurate. After that, the PSO algorithm is employed to solve the optimal control problem about δ in the fixed interval $[t_0, t'_f]$ and the next interval $[t_k, t_f]$. Furthermore, the 4th-order Runge-Kutta numeric integration is employed to evaluate the performance function, while the optimal time t_f can be obtained as long as all the terminal conditions are satisfied.

3. Algorithm Formulation

3.1. LCPSO. The proposed LCPSO is designed as a double iteration scheme: the inner iteration of dual cooperative PSO (CPSO) [14] and the outer iteration of interval analysis. The mathematical formulation of the CPSO algorithm is redesigned in (10) and (11).

$$x_{id}^r(t+1) = x_{id}^r(t) + v_{id}^r(t+1), \quad (10)$$

with $d \in \{1, \dots, N\}$, $l \in \{1, \dots, S\}$, $t < T_{\text{max}}$, $r \in \{1, 2, \dots, R_{\text{max}}\}$,

$$v_{id}^r(t+1) = w_1 \times v_{id}^r(t) + c_1 \times \text{rand}() \times (p_{ld}^r(t) - x_{id}^r(t)) + c_2 \times \text{rand}() \times (p_{gd}^r(t) - x_{id}^r(t)), \quad (11)$$

$$w_1 = w_{\text{min}} + (w_{\text{max}} - w_{\text{min}}) \times \frac{(N_{\text{iter}} - t)}{N_{\text{iter}}},$$

where $x_{id}^r(t)$ is the coordinate of particle i in dimension d at time t as subinterval l at interval iteration step r .

The pseudocode of CPSO algorithm is reported in Pseudocode 1.

```

(1) Input  $P$  (number of cooperation iteration),
(2)    $K$  (number of cooperation iteration),
(3)    $N_{\max}$  (subswarm size),
(4)    $k, l$  and  $r$  (the parameters of interval analysis) are given from outer iteration.
(5) Initialize  $(b_{01}^\alpha, \dots, b_{0N}^\alpha)$  and  $(b_{01}^\delta, \dots, b_{0N}^\delta)$ 
(6) For cooperation iteration  $j \in [1, \dots, P]$ 
(7)   For each subswarm iteration  $t \in [1, \dots, T_{\max}]$ 
(8)     For each particle  $i \in [1, \dots, N_{\max}]$ 
(9)       If  $(j \bmod 2 > 0)$   $\{x_{id}^{rl}(0) = (b_{j1}^\alpha, \dots, b_{jN}^\alpha)\}$ 
(10)      Else  $\{x_{id}^{rl}(0) = (b_{j1}^\delta, \dots, b_{jN}^\delta)\}$ 
(11)      End If
(12)      To run Runge-Kutta numeric integration and evaluate performance function by using Equ. (9).
(13)      If  $(J(x_i^{rl}(t)) < \text{pbestval})$ 
(14)         $\{\text{pbest} = x_i^{rl}(t), \text{pbestval} = J(x_i^{rl}(t))\}$ 
(15)      Else If  $(J(x_i^{rl}(t)) < \text{gbestval})$ 
(16)         $\{\text{gbest} = x_i^{rl}(t), \text{gbestval} = J(x_i^{rl}(t))\}$ 
(17)      End If
(18)      To update the particle by using Equ. (10), Equ. (11).
(19)    End For  $i$ 
(20)  End For  $t$ 
(21)  If  $(j \bmod 2 > 0)$ 
(22)     $\{\text{To update } (b_{j1}^\alpha, \dots, b_{jN}^\alpha) = \text{gbest}\}$ 
(23)  Else  $\{\text{To update } (b_{j1}^\delta, \dots, b_{jN}^\delta) = \text{gbest}\}$ 
(24)  End If
(25) End For  $j$ 

```

PSEUDOCODE 1: Pseudocode of CPSO.

The mathematical formulation of the one-dimensional search method based on interval analysis is designed as

$$k^{rl} = k^{r-1} + (l - 0.5(S + 1))w(k^{r-1}),$$

with $l \in \{1, \dots, S\}$, $r \in \{1, 2, \dots, R_{\max}\}$, (12)

$$w(k^r) = \frac{w(k^{r-1})}{S}.$$

The one-dimensional search algorithm based on the interval analysis is reported in Pseudocode 2.

The midpoint and the initial width of the switching step interval are defined as

$$k^0 = \frac{t_k^0}{T_f} - \frac{(1 - t_k^0/T_f)S}{2}, \quad (13)$$

$$w(k^0) = S \left(1 - \frac{t_k^0}{T_f}\right), \quad (14)$$

where k^{rl} is the ratio between the switching step and the assumption final time in the interval iteration r and the subinterval l . The switch step interval is divided into S subintervals at each interval iteration step r .

The method starts with the initial switch step interval, which is split into multiple subdivisions. Then, those subdivisions are either sent to the solution list which are considered later, or removed from further test list by certain cut-off condition. The above process is repeated by choosing a new

switch step interval until no switch step could be considered or a global optimal point is found.

3.2. Theorem of Boundaries Selection of LCPSO. The Legendre orthogonal polynomials can be generated using Gram-Schmidt orthonormalization [15] in the interval $[-1, 1]$ with the function of the weight $\rho(x) = 1$. $L_N(x)$ represents a Legendre polynomial with degree N . We provide the following transformation in (15) of the independent variable τ of 3-DOF dynamics when $\tau \in [\tau_0, \tau_f]$ to guarantee that $-1 \leq x \leq 1$:

$$x = \frac{2\tau - (\tau_f + \tau_0)}{(\tau_f - \tau_0)}. \quad (15)$$

This way, the continuous function $u(x)$ of variable x can be approximated as

$$u(x) = \sum_{n=0}^N b_n L_n(x). \quad (16)$$

Then, using Legendre polynomial approximation, the original optimal control problem can be transformed into a parameter optimization problem: to find the optimal coefficients $\mathbf{b} = (b_0, b_1, \dots, b_N)$ so as to minimize the fitness in (9) and satisfy the terminal constraints in (8).

Obviously, the appropriate ranges of the coefficients \mathbf{b} can improve the convergence of the optimization procedure and prevent explosion. In our formal studies, a theorem to determine the coefficient of the Legendre orthogonal

```

(1) Input  $T_f, S, R, t_k^0$ ,
(2) Initialize  $t_k^0$  and  $w(k^0)$  by using Equ. (12),
(3) For interval analysis iteration  $r \in [1, \dots, R]$ 
(4)   For each subinterval  $l \in [1, \dots, S]$ 
(5)     To update the switch instant by using Equ. (11).
(6)     To perform CPSO and update  $J(k^{rl})$ 
(7)   End For  $l$ 
(8)     To update  $k^r = \{k^{rl} \mid l : \min(j(k^{rl}))\}$ 
(9)     To update  $t_k^r = k^r T_f$ .
(10) End For  $r$ 
(11) To obtain the best position and the best fitness.

```

PSEUDOCODE 2: Pseudocode of interval analysis.

polynomial is proposed in [16]. However, the statement and proof of the theorem were incomplete and less rigorous. The complete expression of the theorem of how to determine the boundaries of the coefficients of the orthogonal polynomials is proposed and proven.

Theorem 1. Assume that $L_N(x)$ represents an N th degree Legendre polynomial. The continuous function $u(x)$ of variable x can be approximated as in (16). And the range of $u(x)$ belongs to the interval $[U_{\min}, U_{\max}]$ with the variable x presented in (15). Then, the ranges of optimal coefficients $\mathbf{b} = (b_0, b_1, \dots, b_N)$ can be calculated by

$$b_n \in [U_{\min}, U_{\max}], \quad n = 0,$$

$$b_n \in \left[-\left(n + \frac{1}{2}\right)(U_{\max} - U_{\min}) \int_{-1}^1 L_n^+(x) dx, \left(n + \frac{1}{2}\right) \cdot (U_{\max} - U_{\min}) \int_{-1}^1 L_n^+(x) dx \right], \quad n \geq 1,$$

or

$$b_n \in [U_{\min}, U_{\max}], \quad n = 0,$$

$$b_n \in \left[\left(n + \frac{1}{2}\right)(U_{\max} - U_{\min}) \int_{-1}^1 L_n^-(x) dx, -\left(n + \frac{1}{2}\right)(U_{\max} - U_{\min}) \int_{-1}^1 L_n^-(x) dx \right], \quad n \geq 1,$$

where

$$L_n(x) = \begin{cases} 1, & n = 0 \\ \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n, & n = 1, 2, \dots, \end{cases} \quad (19)$$

$$L_n^+(x) = \begin{cases} L_n(x), & L_n(x) \geq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (20)$$

$$L_n^-(x) = \begin{cases} L_n(x), & L_n(x) < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

Proof. According to the assumption, we have the following equations:

$$\begin{aligned} \int_{-1}^1 L_k(x) u(x) dx &= \int_{-1}^1 L_k(x) \sum_{n=0}^N b_n L_n(x) dx \\ &= b_k \int_{-1}^1 L_k(x) L_k(x) dx \\ &\quad + \sum_{n=0, n \neq k}^N b_n \int_{-1}^1 L_k(x) L_n(x) dx. \end{aligned} \quad (22)$$

According to the orthogonal relationship of the Legendre polynomials, we have

$$\int_{-1}^1 L_k(x) L_n(x) dx = \frac{2}{2k+1} \delta_{nk} \quad (23)$$

in which δ_{nk} is called Kronecker Delta and satisfies

$$\delta_{nk} = \begin{cases} 1, & n = k \\ 0, & \text{otherwise.} \end{cases} \quad (24)$$

Substituting (23) and (24) into (22), we have

$$b_k = \left(k + \frac{1}{2}\right) \int_{-1}^1 u(x) L_k(x) dx, \quad \text{with } k \in [0, N]. \quad (25)$$

Assume that $f(x)$ and $g(x)$ are continuous in $[a, b]$ and satisfy $f(x) \leq g(x)$; then, we have

$$\int_a^b f(x) dx \leq \int_a^b g(x) dx \quad (26)$$

except the finite numbers of points.

Since the range of $u(x)$ is $[U_{\min}, U_{\max}]$ and from (20) and (21), we have

$$L_n(x) = L_n^+(x) + L_n^-(x). \quad (27)$$

Then, the following inequations should be achieved:

$$\begin{aligned} U_{\min} L_n^+(x) &\leq L_n^+(x) u(x) \leq U_{\max} L_n^+(x), \\ U_{\max} L_n^-(x) &\leq L_n^-(x) u(x) \leq U_{\min} L_n^-(x). \end{aligned} \quad (28)$$

Hence,

$$\begin{aligned} U_{\max} L_n^-(x) + U_{\min} L_n^+(x) &\leq (L_n^+(x) + L_n^-(x)) u(x), \\ (L_n^+(x) + L_n^-(x)) u(x) &\leq U_{\max} L_n^+(x) + U_{\min} L_n^-(x). \end{aligned} \quad (29)$$

According to (26), (27), and (29), we can obtain

$$\begin{aligned} \int_{-1}^1 L_n(x) u(x) dx &\geq U_{\max} \int_{-1}^1 L_n^-(x) dx \\ &\quad + U_{\min} \int_{-1}^1 L_n^+(x) dx, \\ \int_{-1}^1 L_n(x) u(x) dx &\leq U_{\max} \int_{-1}^1 L_n^+(x) dx \\ &\quad + U_{\min} \int_{-1}^1 L_n^-(x) dx. \end{aligned} \quad (30)$$

Moreover, according to (25) and (30), we have the effective range of b_n :

$$\begin{aligned} b_n \in \left[\frac{2n+1}{2} \left(U_{\min} \int_{-1}^1 L_n^+(x) dx \right. \right. \\ \left. \left. + U_{\max} \int_{-1}^1 L_n^-(x) dx \right), \frac{2n+1}{2} \left(U_{\max} \int_{-1}^1 L_n^+(x) dx \right. \right. \\ \left. \left. + U_{\min} \int_{-1}^1 L_n^-(x) dx \right) \right]. \end{aligned} \quad (31)$$

Therefore, the upper bound and lower bound of coefficients b_n can be defined:

$$\begin{aligned} b_{n\max} &= \frac{2n+1}{2} \left(U_{\max} \int_{-1}^1 L_n^+(x) dx \right. \\ &\quad \left. + U_{\min} \int_{-1}^1 L_n^-(x) dx \right), \\ b_{n\min} &= \frac{2n+1}{2} \left(U_{\min} \int_{-1}^1 L_n^+(x) dx \right. \\ &\quad \left. + U_{\max} \int_{-1}^1 L_n^-(x) dx \right), \end{aligned} \quad (32)$$

with $n = 0, 1, 2, \dots$

The closed form for the orthogonal polynomials is given in (19). Assume that

$$g(x) = (x^2 - 1)^n, \quad n = 1, 2, \dots \quad (33)$$

Then,

$$\begin{aligned} \int_{-1}^1 L_n(x) dx &= \int_{-1}^1 L_n^+(x) dx + \int_{-1}^1 L_n^-(x) dx \\ &= \int_{-1}^1 \frac{1}{2^n n!} g^{(n)}(x) dx = \frac{1}{2^n n!} (x) \Big|_{-1}^1 \\ &= 0. \end{aligned} \quad (34)$$

Therefore,

$$\int_{-1}^1 L_n^+(x) dx = - \int_{-1}^1 L_n^-(x) dx, \quad n = 1, 2, \dots \quad (35)$$

Substituting (35) into (32), we have

$$\begin{aligned} b_{n\max} &= \frac{2n+1}{2} (U_{\max} - U_{\min}) \int_{-1}^1 L_n^+(x) dx, \\ b_{n\min} &= -b_{n\max}, \\ &\quad n \geq 1, \end{aligned} \quad (36)$$

or

$$\begin{aligned} b_{n\max} &= -\frac{2n+1}{2} (U_{\max} - U_{\min}) \int_{-1}^1 L_n^-(x) dx, \\ b_{n\min} &= -b_{n\max}, \\ &\quad n \geq 1. \end{aligned} \quad (37)$$

When $n = 0$, importing $L_0(x) = 1$ and $-1 \leq x \leq 1$ into (32), we have

$$U_{\min} \leq b_0 \leq U_{\max}. \quad (38)$$

The proof is achieved. \square

Consider that the control value $u(x)$ always works in a symmetric feasible interval. Assuming that $U_{\max} > 0$ and $U_{\min} = -U_{\max}$, to simplify the results in practical application, we can obtain

$$\begin{aligned} n = 0, \quad b_0 &\in [U_{\min}, U_{\max}]; \\ n = 1, \quad b_1 &\in [1.5U_{\min}, 1.5U_{\max}]; \\ n = 2, \quad b_2 &\in [1.9245U_{\min}, 1.9245U_{\max}]; \\ n = 3, \quad b_3 &\in [2.275U_{\min}, 2.275U_{\max}]; \\ n = 4, \quad b_4 &\in [2.5793U_{\min}, 2.5793U_{\max}]; \\ n = 5, \quad b_5 &\in [3.8895U_{\min}, 3.8895U_{\max}]. \end{aligned} \quad (39)$$

4. Simulation Results

4.1. Ascent Trajectory Optimization of the X-43 Vehicle. Firstly, we use the model of the X-43 vehicle [13] as the test case. Particularly, the vector of the initial states is $[r_0, \theta_0, \varphi_0, v_0, \gamma_0, m_0] = [6398000, 120, 0, 1200, 0, 1000]$, while the terminal states are $[r_f, v_f, \gamma_f] = [6408000, 1820, 0]$. The tolerance of state variables in the terminal point is $[\Delta r, \Delta v, \Delta \gamma] = [\pm 1000, \pm 50, \pm 1.5]$. Some other experiment parameters are as follows: the number of LCPSO iterations that occurred $T = 35$, the population size of a single particle swarm $N = 40$, the dimension of the particle $D = 6$, the number of swarm iterations of the cooperated particle $P = 4$, the acceleration coefficients $c_1 = c_2 = 2$, and the range of the inertia weight $w_{\max} = 0.9$ and $w_{\min} = 0.4$. In addition, the weights of the performance function are

TABLE 1: Results after 1 time numerical simulation.

Iteration time t	Switching step ratio k	Corresponding step t_k	Optimal cost J
1	0.450	90.0	320.849
1	0.550	110.0	314.069
1	0.650	130.0	291.451
1	0.750	150.0	433.650
1	0.850	170.0	491.392
2	0.610	122.0	293.675
2	0.630	126.0	254.625
2	0.650	130.0	275.729
2	0.670	134.0	294.822
2	0.690	138.0	348.983
3	0.622	124.4	285.098
3	0.626	125.2	259.907
3	0.630	126.0	254.625
3	0.634	126.8	262.411
3	0.638	127.6	278.003

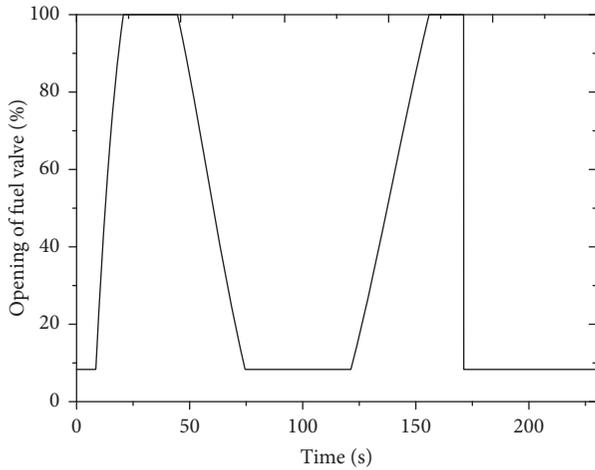


FIGURE 1: The profile of opening of fuel valve.

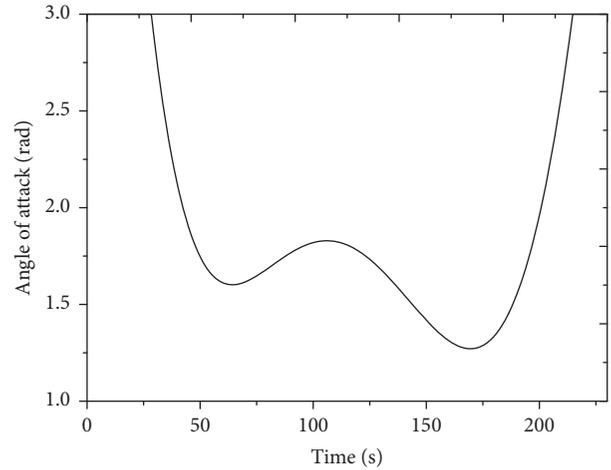


FIGURE 2: The profile of the angle of attack.

$[k_1, k_2, k_3] = [10^9, 10^4, 10^3]$, and the parameters of interval analysis iteration are $[t'_f, t_k, S, R] = [1, 0.9, 5, 3]$.

In Table 1, during the optimal processing, five subintervals are introduced and the optimal costs vary with the switch steps in each iteration step of interval analysis, and the results converged at the global optimal point only through three steps of iteration. It takes about 580 seconds under the hardware environment of Intel Core i3 3220 CPU and DDR4G 1600 MHz memory and the software environment of Windows 7 operating system with VC++ 6.0 compiling environment.

The optimal flight path angle profile is shown in Figure 1, with the corresponding optimal control variables profile in Figures 2 and 3.

In Table 2, according to the ratios of the feasible results, the best and the mean results of 200 times simulations using interval analysis of LCPSO with different coefficients' ranges are presented. Apparently, the proposed method with the

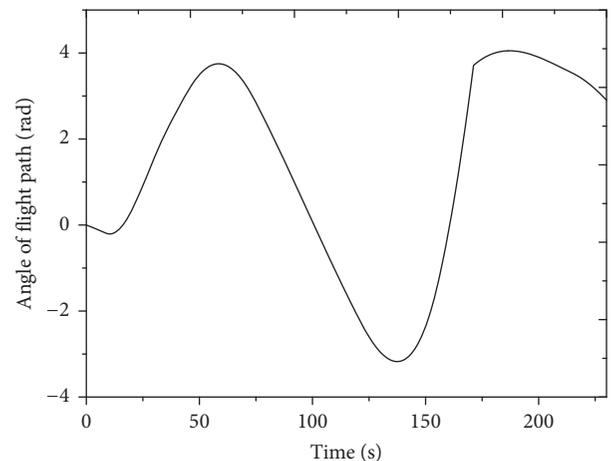


FIGURE 3: The profile of the angle of flight path.

TABLE 2: Coefficient range selection results.

	Standard range	0.5 times range	2 times range	5 times range
Ratio of feasible results	72.0	59.1	63.5	24.5
Least consumption of fuel	51.8%	55.9%	53.6%	56.4%
Mean consumption of fuel	63.0%	61.1%	64.5%	70.5%

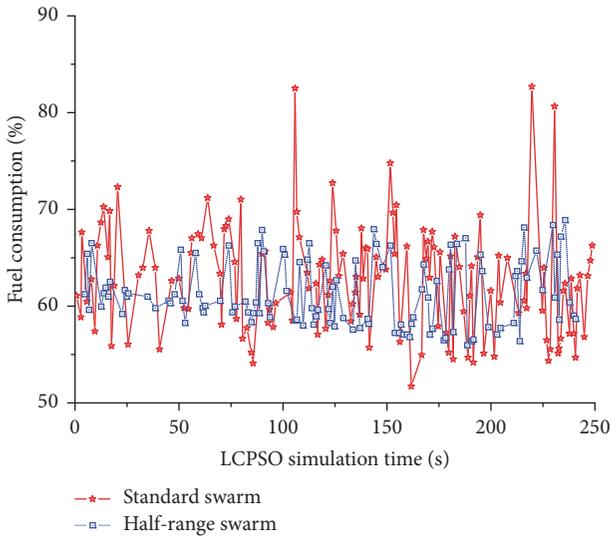


FIGURE 4: Result distribution with standard and half times range.

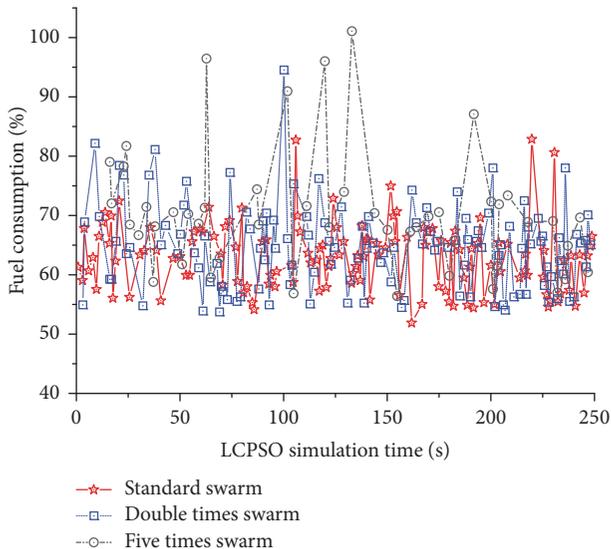


FIGURE 5: Result distribution with standard, double times, and five times range.

coefficients' ranges selection has the best performance, while all state variables meet the state terminal constraints with higher precision. Furthermore, the feasible results distribution with standard range and half-range selection is shown in Figure 4, while the feasible results distributions with

standard, two, and five times range selection are displayed in Figure 5. From the comparison, we can find that if the optimal solutions existed in this reductive area, the reduction of the search range to its half will probably decrease the search complexity. However, if the performance index function corresponding to the reductive area is of bad quality, the optimized solution is likely to be lost. On the other hand, the multifold coefficient will provide the unwanted search space, which will increase the complexity and reduce efficiency of the optimization.

4.2. Earth-Mars Transfer Problem. The orbit transfer problem is another kind of hotspot issues on trajectory optimization research [17]. Here, we use the proposed LCPSO method to solve the Earth-Mars trajectory transfer problem and compare it with the traditional DE and GA method.

The Earth-Mars transfer orbit can be divided into three stages: geocentric escape section, heliocentric transition section, and areosynchronous capture section. Each segment has different constraints. In the geocentric escape section, the initial state constraints are $[r_0, v_{r0}, v_{\theta0}] = [6.6107R_E, 0, 0.38893R_E/TU_E]$ and the terminal state constraint is $r_f = 145R_E$. On the other hand, the constraints required in the third segment are the terminal radius ρ_f and the terminal speeds $v_{\rho f}$ and $v_{\delta f}$, which satisfy $[\rho_f, v_{\rho f}, v_{\delta f}] = [6.0236R_M, 0, 0.40745R_M/TU_M]$. And the initial radius is the radius of the sphere of influence for Mars, $\rho_0 = 170R_M$.

The LCPSO algorithm parameters are set as follows: the population size $m = 20$ and the maximum number of iterations $n = 200$. Meanwhile, the parameters of DE and GA algorithm are set as follows: the population size $m = 20$, the maximum number of iterations $n = 200$, the cross probability $p_c = 0.4$, and the mutation probability $p_m = 0.6$. The optimization variables contain the thrust angle of the aircraft, the initial and terminal orbital angles, and the time of the whole three-segment orbits flight. Currently, the range of the thrust angle and the initial/terminal orbital angles of the vehicle is set in $[0, \pi]$. The problem is to transfer the orbit within the shortest time to minimize the fuel consumption, so the fitness function can be described as $J = t_1 + t_2 + t_3$, where t_1 , t_2 , and t_3 are the transfer periods of three sections, respectively. Figure 6 shows the best fitness process using LCPSO, DE, and GA algorithms. It is shown that the LCPSO has a faster speed of convergence and higher accuracy than DE and GA algorithms.

The final resulting candidate solutions by LCPSO, DE, and GA are shown in Figures 7–9. Because the intelligent algorithm obtains the feasible solution to meet the accuracy requirements, the state curves of any two algorithms are not

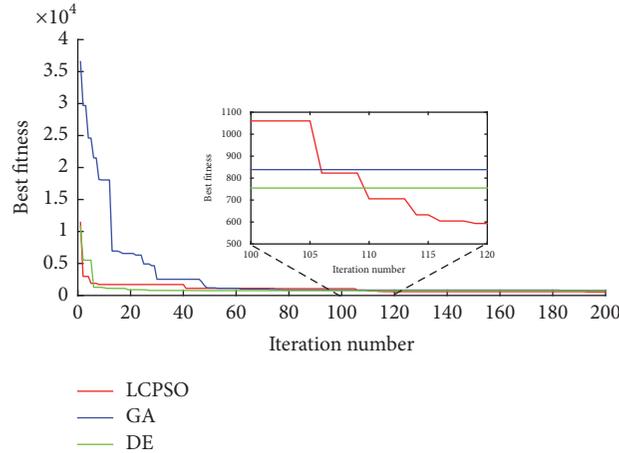


FIGURE 6: Convergence processes of the fitness functions.

TABLE 3: Final times for the geocentric, heliocentric, and areocentric segments.

Segment	Final time (days)		
	LCPSO	GA	DE
1 (geocentric)	37.1	34.6	34.5
2 (heliocentric)	171.4	182.4	174.7
3 (areocentric)	19.3	20.3	19.6
Whole flight time	227.8	237.3	228.8

necessarily completely consistent. But both of them satisfy the constraints and accuracy requirements. For instance, as shown in Figure 7, although the final radius of the geocentric orbit is about $145R_E$ by LCPSO, $145.1R_E$ by GA, and $143R_E$ by DE, the orbital radius and the velocity are all within the permitted constrained ranges. And in this case, the optimization result of the final geocentric orbital radius by LCPSO is closer to the constraint requirements. Hence, the LCPSO exhibits better efficiencies on the constraints requirements than the other two algorithms. Moreover, the results of the whole transfer time by different optimization algorithms are shown in Table 3. It can be found that, in the case of satisfying constraints, the optimal trajectory by LCPSO will spend the shortest flight time among the three algorithms, which has the best performance.

5. Conclusion

In this paper, a novel interval analysis based Legendre cooperative PSO algorithm is proposed and applied to solve the trajectory optimization problems. The Legendre orthogonal polynomial approximation is synthesized with the dual cooperative particle swarms to transfer the arbitrary final time optimal problem into a two-point boundary value problem with fixed terminal one. Then, a fast one-dimensional interval search method is provided in each selected interval to reduce the search space of the particles in the iterations. Furthermore, a theorem that determines the range of the

parameters of the Legendre polynomial is investigated, and the problem can be solved to get closer to the global optimal solution.

Lastly, in numeral simulation, the results demonstrate that the LCPSO algorithm can solve the unsmooth trajectory optimization problem of X-43 effectively and obtain a smooth control variable. And the appropriate range of parameter values will significantly reduce the complexity of the optimization search. Moreover, in the solution to the orbit transfer problem, the comparisons with the existing GA and DE algorithms represent the notion that the proposed LCPSO method has better performance in the speed of convergence, final accuracy, and constraints satisfaction.

The flaw of the proposed LCPSO algorithms is that the parameters of the Legendre polynomial are sensitive in adjustment. Nevertheless, in the trajectory optimization problem of HV, the variety of parameters is a relatively gentle process because of the dynamic characters of HV, and the parameters are adjusted less frequently in iterations. Normally, the parameters will remain the same after they are adjusted once before. The optimization between parameter sensitivity and optimal solution will be studied to make the method more practical in engineering in our future work.

Nomenclature and Abbreviations

CPSO:	Cooperative particle swarm optimization
DE:	Differential evolution
DOF:	Degree of freedom
FER:	Fuel expendable ratio
HV:	Hypersonic vehicle
GA:	Genetic algorithm
LCPSO:	Legendre cooperative particle swarm optimization
PSO:	Particle swarm optimization.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

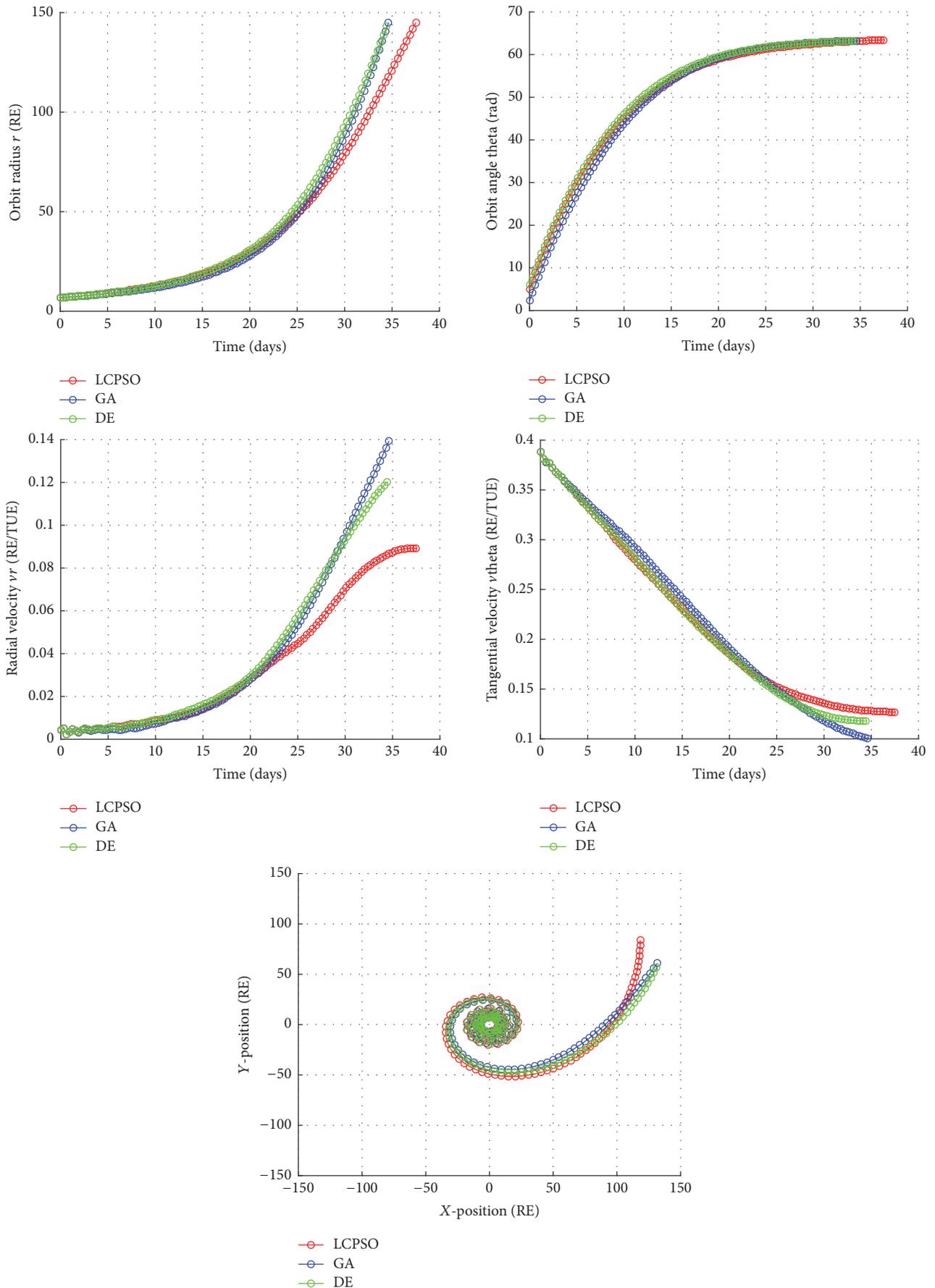


FIGURE 7: State variables in geocentric segment.

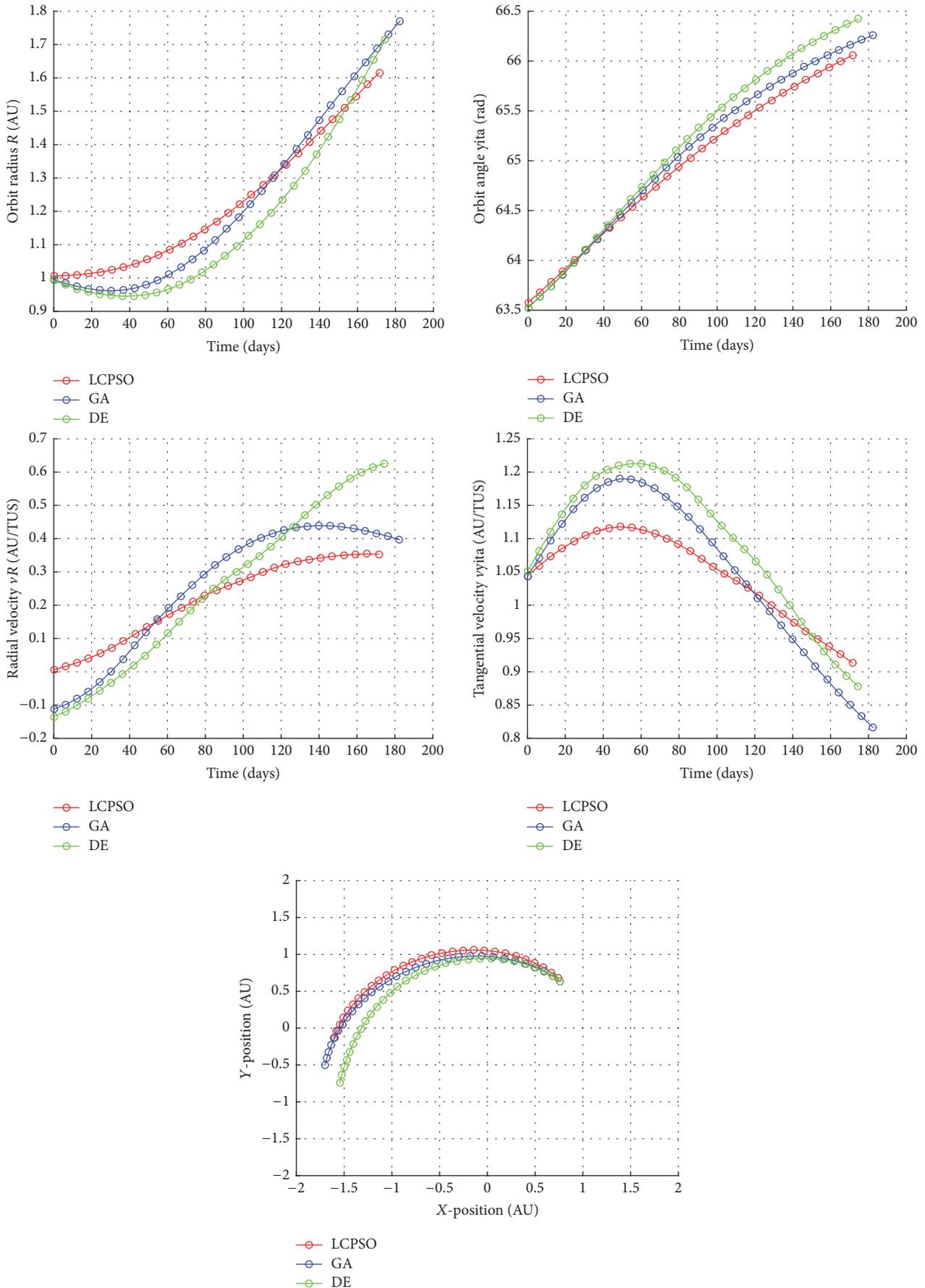


FIGURE 8: State variables in heliocentric segment.

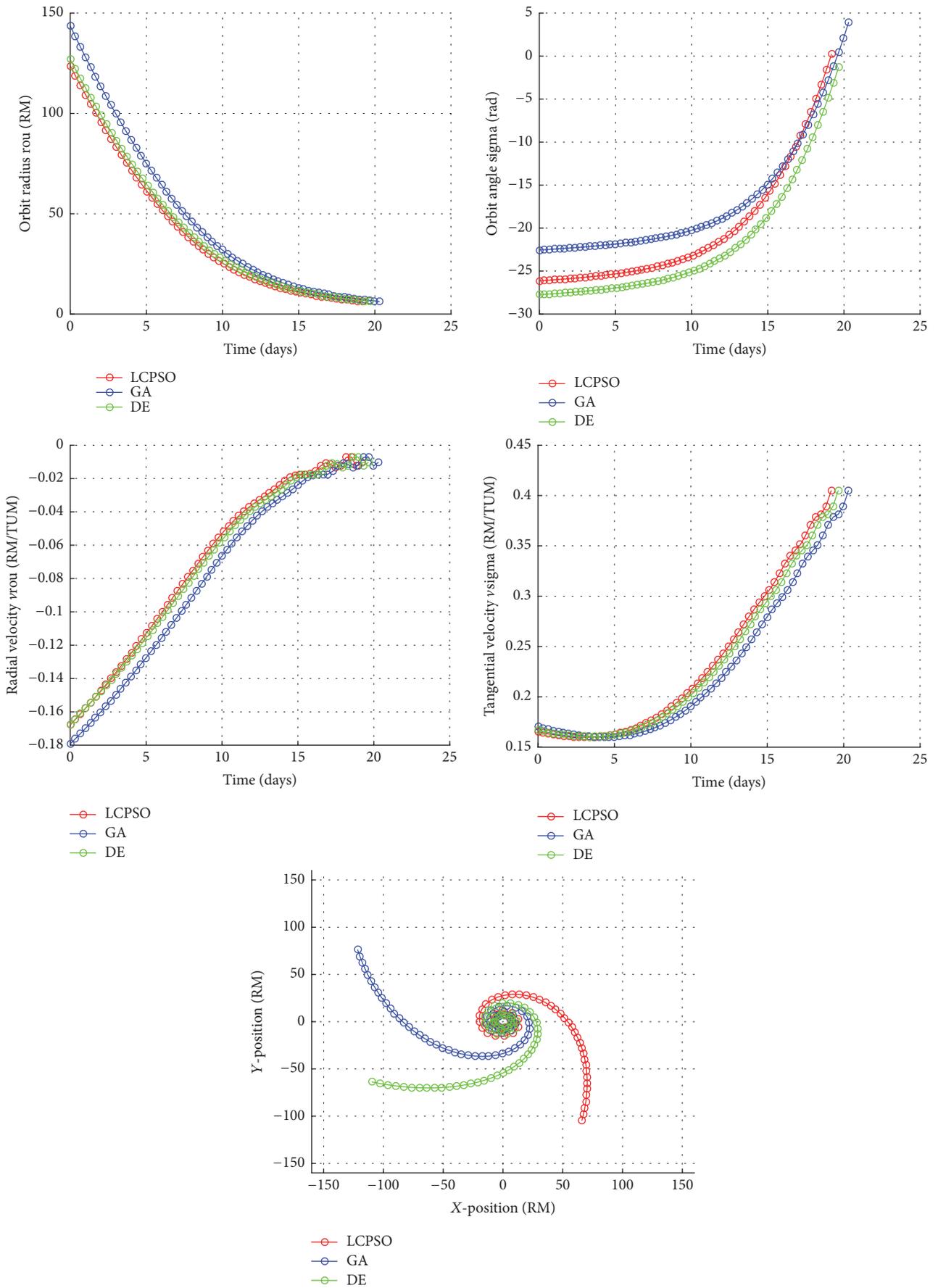


FIGURE 9: State variables in areocentric segment.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grants nos. 61573161 and 61473124).

References

- [1] B. Xu and Z. Shi, "An overview on flight dynamics and control approaches for hypersonic vehicles," *Science China Information Sciences*, vol. 58, no. 7, 070201, 19 pages, 2015.
- [2] H. Luo, Y. Niu, M. Zhu, X. Hu, and H. Ma, "Optimization of Pesticide Spraying Tasks via Multi-UAVs Using Genetic Algorithm," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–16, 2017.
- [3] M. Vasile, E. Minisci, and M. Locatelli, "An inflationary differential evolution algorithm for space trajectory optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 267–281, 2011.
- [4] Q. Lin, S. Liu, Q. Zhu et al., "Particle Swarm Optimization With a Balanceable Fitness Estimation for Many-Objective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 32–46, 2018.
- [5] J. Huang, J. Qian, L. Liu, Y. Wang, C. Xiong, and S. Ri, "Echo state network based predictive control with particle swarm optimization for pneumatic muscle actuator," *Journal of The Franklin Institute*, vol. 353, no. 12, pp. 2761–2782, 2016.
- [6] F. Zhao, Y. Liu, K. Huo, and Z. Zhang, "Radar Target Classification Using an Evolutionary Extreme Learning Machine Based on Improved Quantum-Behaved Particle Swarm Optimization," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–13, 2017.
- [7] M. Mohammadi and N. Ghadimi, "Optimal location and optimized parameters for robust power system stabilizer using honeybee mating optimization," *Complexity*, vol. 21, no. 1, pp. 242–258, 2015.
- [8] T. Mahto and V. Mukherjee, "Frequency stabilisation of a hybrid two-area power system by a novel quasi-oppositional harmony search algorithm," *IET Generation, Transmission & Distribution*, vol. 9, no. 15, pp. 2167–2179, 2015.
- [9] Z. Wang, H. Xing, T. Li, Y. Yang, R. Qu, and Y. Pan, "A Modified Ant Colony Optimization Algorithm for Network Coding Resource Minimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 325–342, 2016.
- [10] J. Pekonen, J. Nam, J. O. Smith, and V. Valimaki, "Optimized Polynomial Spline Basis Function Design for Quasi-Band-limited Classical Waveform Synthesis," *IEEE Signal Processing Letters*, vol. 19, no. 3, pp. 159–162, 2012.
- [11] S. M. Abd-Elazim and E. S. Ali, "A hybrid particles swarm optimization and bacterial foraging for power system stability enhancement," *Complexity*, vol. 21, no. 2, pp. 245–255, 2015.
- [12] D. Zhang, L. Liu, and Y. Wang, "On-line Ascent Phase Trajectory Optimal Guidance Algorithm based on Pseudo-spectral Method and Sensitivity Updates," *Journal of Navigation*, vol. 68, no. 6, pp. 1056–1074, 2015.
- [13] M. A. Bolender and D. B. Doman, "Nonlinear longitudinal dynamical model of an air-breathing hypersonic vehicle," *Journal of Spacecraft and Rockets*, vol. 44, no. 2, pp. 374–387, 2007.
- [14] C. Knievel and P. A. Hoehner, "On particle swarm optimization for MIMO channel estimation," *Journal of Electrical and Computer Engineering*, Article ID 614384, 2012.
- [15] M. Dadkhah, M. H. Farahi, and A. Heydari, "Optimal control of a class of non-linear time-delay systems via hybrid functions," *IMA Journal of Mathematical Control and Information*, vol. 34, no. 1, pp. 255–270, 2017.
- [16] F. Xie, Y. Yang, S. Chang, and Y. Wang, "Improving strategies on PSO for suborbit launch vehicle trajectory optimization," in *Proceedings of the 4th International Workshop on Advanced Computational Intelligence, IWACI 2011*, pp. 113–119, China, October 2011.
- [17] X.-Y. Gao, K. L. Teo, and G.-R. Duan, "Non-fragile guaranteed cost control for robust spacecraft orbit transfer with small thrust," *IMA Journal of Mathematical Control and Information*, vol. 28, no. 4, pp. 507–524, 2011.

Research Article

Extension of the Multi-TP Model Transformation to Functions with Different Numbers of Variables

Péter Baranyi 

Széchenyi István University, Győr, Hungary

Correspondence should be addressed to Péter Baranyi; prof.peter.baranyi@gmail.com

Received 18 September 2017; Accepted 29 January 2018; Published 19 March 2018

Academic Editor: Kevin Wong

Copyright © 2018 Péter Baranyi. 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 tensor product (TP) model transformation defines and numerically reconstructs the Higher-Order Singular Value Decomposition (HOSVD) of functions. It plays the same role with respect to functions as HOSVD does for tensors (and SVD for matrices). The need for certain advantageous features, such as rank/complexity reduction, trade-offs between complexity and accuracy, and a manipulation power representative of the TP form, has motivated novel concepts in TS fuzzy model based modelling and control. The latest extensions of the TP model transformation, called the multi- and generalised TP model transformations, are applicable to a set functions where the dimensionality of the outputs of the functions may differ, but there is a strict limitation on the dimensionality of their inputs, which must be the same. The paper proposes an extended version that is applicable to a set of functions where both the input and output dimensionalities of the functions may differ. This makes it possible to transform complete multicomponent systems to TS fuzzy models along with the above-mentioned advantages.

1. Introduction

The appearance of the Singular Value Decomposition (SVD) was one of the largest breakthroughs in matrix algebra [1]. Its applicability was extended to tensors in the form of the Higher-Order SVD [2] around 2000. Recently, a further extension of the SVD and HOSVD concept, known as the tensor product (TP) Model Transformation, was proposed for functions in control theory [3]. A comprehensive overview is given in [4]. Various extensions of the TP model transformation such as the bilinear-, pseudo-, multi-, and generalised TP model transformation, as well as the concept of HOSVD canonical form of TS fuzzy or TP models, were proposed in [4–7], with a special focus on TS fuzzy models in [8]. The approximation power of the TP model transformation applied to TS fuzzy models is investigated in [9].

The above-mentioned extensions and variations of the TP model transformation were primarily applied to fuzzy model complexity reduction [10, 11] and in the widely used TS fuzzy model based PDC (Parallel Distributed Compensation) control theories [12–14]. But also, in general, it has been applied to polytopic model, TP/TS fuzzy model, and LMI (Linear Matrix Inequality [15]) based control theories. The most important features of the TP model transformation

are guaranteed by the key transformation step whereby a numerically reconstructed HOSVD structure is determined. Key features of the transformation are as follows:

- (i) It is executable on models given by equations or soft computing based representations, such as fuzzy rules or neural networks or other black-box models. The only requirement is that the model must provide an output for each input (at least on a discrete scale, see Section 4, Step 1).
- (ii) It will find the minimal complexity, namely, the minimal number of rules of the TS fuzzy model. If further complexity reduction is required, it provides one of the best trade-offs between the number of rules and approximation error.
- (iii) It works like a principle component analysis, in that it determines the order of the components/fuzzy rules according to their importance.
- (iv) It is capable of deriving the antecedent fuzzy sets according to various constraints. For instance, it can be used to define different convex hulls, a capability which has recently been shown to play an important role in control theory.

- (v) It is capable of transforming the given model to predefined antecedent fuzzy sets (pseudo-TP model transformation)
- (vi) It is capable of transforming a set of models simultaneously, while common antecedent fuzzy sets are derived for all models.

Based on the above, various theories and applications have emerged using the TP model transformation. Further computational improvements were proposed in [16, 17]. It has been proved in [5, 18–20] that LMI based control design theories are very sensitive for convex hulls defined by consequents (vertices) of TS fuzzy models. Thus, the convex hull manipulation capability of the TP model transformation is an important and necessary step in LMI based control design. Very effective convex hull manipulation methods were incorporated into the TP model transformation in [21–23]. Further useful control approaches and applications were published in the field of control theory [24–41]. Many powerful approaches are published on the field of sliding mode control in [29, 42, 43]. In physiological control the usability of TP model transformation has been demonstrated as well [44–49]. Various further theories and applications are studied in [50–87].

One of the key advantages of the TP model transformation is that is capable of finding the minimal complexity of all components of the system and guarantees the same antecedent system for all components. This is a very typical requirement in design or stability verification methodologies, that is, the model, controller, and observer need to have the same antecedent system, hence, convex representation. Therefore, the simultaneous manipulation of the components with the multi-TP model transformation or the generalised TP model transformation (that combines all variants of the TP model transformation) yields further possibilities for control performance optimisation [18–20].

Despite the above advantages, a crucial limitation of the generalised TP model transformation is that it can only be applied to a set of systems which have the same number of inputs. For instance, consider four different systems given with different representations, as shown in Figure 1. S1 is a fuzzy logic model; S2 is neural network; S3 is given by an equation; and S4 is a black-box model. All of these models have the same inputs but may have different sized output tensors. The multi-TP model transformation is capable of simultaneously transforming all systems to TP or TS fuzzy model form, such that the same antecedent sets are defined on the inputs. The generalised TP model transformation can also transform to predefined antecedent fuzzy sets.

A further generalisation proposed in this paper can be applied to systems like in the example given in Figure 2. Here each system may be given by different representations (like in the above case) but may also have different numbers of inputs. The transformation can simultaneously convert all of the systems to TS fuzzy model form, such that the antecedent fuzzy sets will either be the same or assume a predefined structure. From all other perspectives, the proposed TP model transformation inherits all of the advantageous features of the previous TP-based approaches.

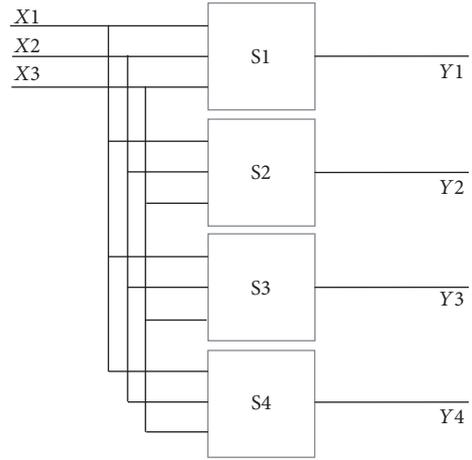


FIGURE 1: Multi TP model transformation.

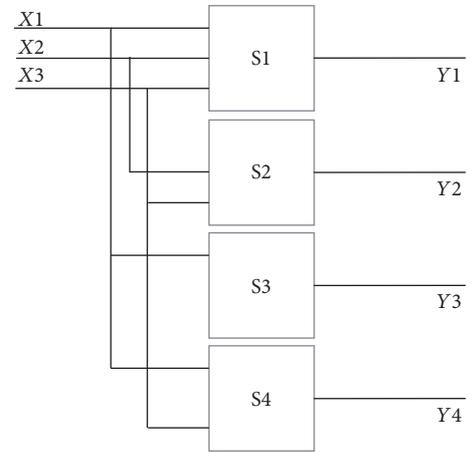


FIGURE 2: Proposed extension of the TP model transformation.

Recently proposed SOS-type (Sum-of-squares) TS fuzzy LPV models are also widely applied in fuzzy control theories [88, 89]. The further extension of the TP model transformation to such systems is highly welcome in future works.

2. Notation and Concepts

2.1. *Notation.* The following notations are used in the paper:

- (i) Scalar: a is scalar.
- (ii) Vector: \mathbf{a} contains elements a_i .
- (iii) Matrix: \mathbf{A} contains elements $a_{i,j}$.
- (iv) Tensor: \mathcal{A} contains elements $a_{i,j,k,\dots}$.
- (v) Set: $A = \{a, b, c, \dots\}$, for example, $a \in A$.
- (vi) Index i : the upper bounds of the indices are denoted by the uppercase letter, for example, I .
- (vii) Index $i \in I$ denotes that index i takes the elements of set $I \subseteq \{1, 2, \dots, I\} \subset \mathbb{N}$, respectively. $I : \{1, 2, \dots, I\}$ is understood as per default.
- (viii) Interval: $\omega = [\omega_{\min}, \omega_{\max}]$.

- (ix) Space: $\Omega : \omega_1 \times \omega_2 \times \dots \times \omega_N$ is an N dimensional hypercube.
- (x) $\mathbf{x} \in \Omega$ expresses the fact that vector \mathbf{x} is within the space Ω . The dimensions of \mathbf{x} and Ω are the same.
- (xi) \sqsubset denotes a dimensionality reduced subset in general as follows:
- In the case of spaces: $\Theta \sqsubset \Omega$ states that Θ is a hypercube with the same sized intervals as Ω , but has a smaller number of dimensions.
 - In the case of vectors: $\mathbf{a} \sqsubset \mathbf{b}$, where $\mathbf{a} \in \Theta \subset \mathbb{R}^N$ and $\mathbf{b} \in \Omega \subset \mathbb{R}^M$ means that $N < M$ and $\Theta \sqsubset \Omega$.
 - In the case of tensors: $\mathcal{A} \sqsubset \mathcal{B}$ means for instance that \mathcal{A} is obtained by deleting complete dimensions from tensor \mathcal{B} .
- (xii) Grid: $G : \mathbf{g}_1 \times \mathbf{g}_2 \times \dots \times \mathbf{g}_N$ is a rectangular hyper grid (tensor), where $\mathbf{g}_n = [g_{n,1} < g_{n,2} < \dots < g_{n,M_n}]$ defines the locations of the M_n different grid points in increasing order.
- (xiii) Pair (Ω, G) : space $\Omega \in \mathbb{R}^N$ and grid G are in a pair, meaning that $\forall n \in \mathbb{N} : g_{n,1} = \omega_{n,\min}$ and $g_{n,M_n} = \omega_{n,\max}$.
- (xiv) Discretised function $\mathcal{F}^{D(\Omega,G)}$ of $f(\mathbf{x})$ denotes the sampling of $f(\mathbf{x})$ over pair (Ω, G) . Thus, it is a tensor with the size of $M_1 \times M_2 \times \dots \times M_N$ and entries:
- $$f_{m_1, m_2, \dots, m_N} = f([g_{1, m_1} \ g_{2, m_2} \ \dots \ g_{N, m_N}]). \quad (1)$$
- (xv) $\mathcal{S} \boxtimes_N \mathbf{U}_n$ is the tensor product (TP); for details, refer to [4, 5, 8]. A slight difference in notation here is that N under the tensor product operation \boxtimes is only a set numbers $n \in \mathbb{N}$ to which the product should be applied.
- (xvi) $f(\mathbf{x}) = \mathcal{S} \boxtimes_N \mathbf{w}_n(x_n)$ represents the TP function, $\mathbf{x} \in \mathbb{R}^N$, where $\mathbf{w}_n(x_n) = [w_{n,1}(x_n) \ w_{n,2}(x_n) \ \dots \ w_{n,I_n}(x_n)]$ is called the weighting function system.
- (xvii) Types of the weighting functions are as follows:

- SN: sum normalised
- NN: nonnegativeness
- NO: normalised
- CNO: close to normalised
- RNO: relaxed normalised
- INO: inverse normalised
- IRNO: inverse relaxed normalised.

For further details, refer to [4, 5].

3. The Proposed TP Model Transformation

Assume that a set of functions is given as $\mathcal{Y}_l = f_l(\mathbf{x}_l)$, $\mathbf{x}_l \sqsubseteq \mathbf{x} \in \Omega$; thus $\mathbf{x}_l \in \Omega_l \sqsubseteq \Omega \subset \mathbb{R}^N$, $\mathbf{x}_l \in \mathbb{R}^{N_l}$. The output tensor \mathcal{Y}_l of each function $f_l(\mathbf{x}_l)$ may differ in the number of dimensions and its size as $\mathcal{Y}_l \in \mathbb{R}^{O_{1,l} \times O_{2,l} \times \dots \times O_{K_l,l}}$, where K_l denotes the number of dimensions of the output and O_k denotes the number of elements in dimension k .

The goal of the TP model transformation is to transform $\forall l : f_l(\mathbf{x}_l)$ into TP function form as

$$\mathcal{Y}_l = f_l(\mathbf{x}_l) = \mathcal{S}_l \boxtimes_{N_l} \mathbf{w}_{l,n}(x_n) \quad (2)$$

under the following constraints given on the weighting functions.

(i) *Unified Constraints for $\forall l : f_l(\mathbf{x}_l)$.* All resulting TP functions will have the same weighting function system on each dimension defined by the set $V \subseteq \mathbb{N}$ (obviously, if the function has that input dimension):

- Weighting function systems $\mathbf{w}_a(x_a)$, $a \in A \subseteq V$, are predefined.
- Weighting function systems $\mathbf{w}_b(x_b)$, $b \in B \subseteq V$ will be derived by the transformation; only their types are predefined (i.e., SN, NN, NO, CNO, RNO, INO, and IRNO). Further the number of the weighting functions are minimised.

(ii) *Different Constraints for Each $f_l(\mathbf{x}_l)$.* The resulting TP functions have different weighting functions on dimensions $Z \subseteq \mathbb{N}$:

- Weighting function systems $\mathbf{w}_{l,c}(x_c)$, $c \in C \subseteq Z$ are predefined of each $f_l(\mathbf{x}_l)$.
- The types (i.e., SN, NN, NO, CNO, RNL, INO, and IRNO) of the weighting function systems $\mathbf{w}_{l,d}(x_d)$ are predefined for dimensions $d \in D \subseteq Z$ of each $f_l(\mathbf{x}_l)$.

Thus, (2) can be given as follows:

$$\begin{aligned} \mathcal{Y}_l &= f_l(\mathbf{x}_l) \\ &= \mathcal{S}_l \boxtimes_A \mathbf{w}_a(x_a) \boxtimes_B \mathbf{w}_b(x_b) \boxtimes_C \mathbf{w}_{l,c}(x_c) \boxtimes_D \mathbf{w}_{l,d}(x_d). \end{aligned} \quad (3)$$

4. The Computation of the Proposed TP Model Transformation

Step 1 (discretisation).

- Discretisation of all $f_l(\mathbf{x}_l)$ results in tensor $\mathcal{F}_l^{D(\Omega_l, G_l)}$, ($G_l \sqsubseteq G$). The size of $\mathcal{F}_l^{D(\Omega_l, G_l)}$ in dimension $n \in N_l$ is M_n .
- Discretise the predefined weighting functions over the dimensions of Ω :

$$\begin{aligned} \mathcal{W}_a^{D(\omega_a, \mathbf{g}_a)}, \\ \mathcal{W}_{l,c}^{D(\omega_c, \mathbf{g}_c)}. \end{aligned} \quad (4)$$

Remark 1. This step is executed in the same way as in the case of the original TP model transformation; see [4, 5, 8].

Step 2 (defining TP structures). Execute the following steps in each dimension $n \in \mathbb{N}$:

- Lay out tensors $\mathcal{F}_l^{D(\Omega_l, G_l)}$ in dimensions n if vector \mathbf{x}_l has the following dimension:

$$\mathbf{H}_l = \left(\mathcal{F}_l^{D(\Omega_l, G_l)} \right)_{(n)}. \quad (5)$$

(ii) If $n \in B$ then create

$$\mathbf{T}_n = [\mathbf{H}_1 \ \mathbf{H}_2 \ \cdots \ \mathbf{H}_L]. \quad (6)$$

Execute SVD on \mathbf{T}_n and SN, NN, NO, CNO and complexity trade-off by discarding singular values in the same way as in the original TP model transformation, which results in

$$\mathbf{T}_n = \mathbf{U}_n \mathbf{D}_n \mathbf{V}_n^T. \quad (7)$$

As a matter of fact, if nonzero singular values are discarded then it is only an approximation. Let

$$\mathcal{W}_n^{D(\omega_n, \mathbf{g}_n)} = \mathbf{U}_n. \quad (8)$$

(iii) If $n \in D$ then execute SVD on \mathbf{H}_l as

$$\mathbf{H}_l = \mathbf{U}_l \mathbf{D}_l \mathbf{V}_l^T; \quad (9)$$

and, according to the conditions, execute SN, NN, NO, CNO, and complexity trade-off by discarding singular values in the same way as in the original TP model transformation:

$$\mathbf{H}_l = \mathbf{U}'_l \mathbf{D}'_l \mathbf{V}'_l{}^T. \quad (10)$$

Again, if nonzero singular values are discarded then it is only an approximation. Let

$$\mathcal{W}_{l,n}^{D(\omega_n, \mathbf{g}_n)} = \mathbf{U}'_l. \quad (11)$$

(iv) Finally,

$$\mathcal{S}_l = \mathcal{F}_l^{D(\Omega, G)} \boxtimes_{n \in A \cup B} \mathbf{K}_n \boxtimes_{n \in C \cup D} \mathbf{K}_{l,n}, \quad (12)$$

where

$$\begin{aligned} \mathbf{K}_n &= \left(\mathcal{W}_n^{D(\omega_n, \mathbf{g}_n)} \right)^+, \\ \mathbf{K}_{l,n} &= \left(\mathcal{W}_{l,n}^{D(\omega_n, \mathbf{g}_n)} \right)^+, \end{aligned} \quad (13)$$

where $(\cdot)^+$ denotes the pseudoinverse.

Step 3 (reconstruction of the weighting functions). This step is the same as in the multi-TP model transformation [4, 5, 8]. Having the result of the above steps, $\mathcal{F}_l^{D(\Omega, G)}$ and \mathcal{S}_l , we can recalculate the weighting functions at any point. We may calculate the first two steps over a grid, which is not too dense, but calculate the weighting function over a very dense grid (as suggested in [5]), and then construct piecewise linear functions. As a result we have $\mathbf{w}_v(x_v)$ and $\mathbf{w}_{l,z}(x_z)$.

Then we achieved the goal. We have the TP model form of all functions with the given constraints:

$$f_l(\mathbf{x}_l) = \mathcal{S}_l \boxtimes_{N_l} \mathbf{w}_{l,n}(x_n), \quad (14)$$

or

$$f_l(\mathbf{x}_l) \approx \mathcal{S}_l \boxtimes_{N_l} \mathbf{w}_{l,n}(x_n), \quad (15)$$

if a complexity trade-off is executed (nonzero singular values are discarded), where

$$\forall l : \mathbf{w}_{l,a}(\mathbf{x}_l) = \mathbf{w}_a(\mathbf{x}_l),$$

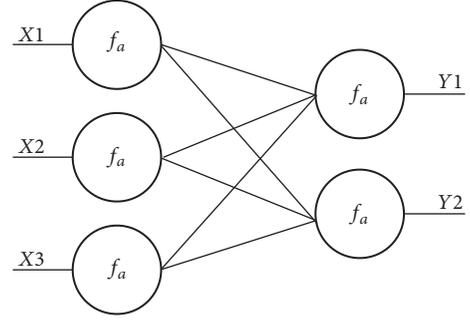


FIGURE 3: System 3: neural network.

$$\forall l : \mathbf{w}_{l,b}(\mathbf{x}_l) = \mathbf{w}_b(\mathbf{x}_l). \quad (16)$$

Or in other words,

$$f_l(\mathbf{x}_l) = \mathcal{S}_l \boxtimes_A \mathbf{w}_a(\mathbf{x}_a) \boxtimes_B \mathbf{w}_b(\mathbf{x}_b) \boxtimes_C \mathbf{w}_{l,c}(\mathbf{x}_l) \boxtimes_D \mathbf{w}_{l,d}(\mathbf{x}_l). \quad (17)$$

Remark 2. The convex hull manipulation and the complexity trade-off are done in the second step. Therefore the approximation accuracy is controlled here by the discarded nonzero singular values. However, the discarded nonzero singular values lead to approximation error. If the given weighting function system is not sufficient (i.e., the number of the weighting functions is less than the rank of that dimension) then we arrive at an approximation only. The use of the pseudoinverse guarantees, however, that it will be the best approximation.

5. Example

5.1. The System. Consider a multicomponent system with input vector $\mathbf{x} = [x_1 \ x_2 \ x_3]$, where $\mathbf{x} \in \Omega = \omega_1 \times \omega_2 \times \omega_3 \in \mathbb{R}^3$ and $\omega_n = [0, 1]$. The system has four subsystems, $l \in L : \{1, \dots, 4\}$ as shown in Figure 1.

System 3. In order to have a systematic notation, we denote the input vector of System 3 as $\mathbf{x}_1 = \mathbf{x}$ that is $\mathbf{x}_1 \in \Omega_1 = \Omega$. It is a neural network; see Figure 3:

$$\begin{aligned} y_1 &= f_a \left(\sum_{i=1}^3 f_a(x_i) b_{1,i} \right), \\ y_2 &= f_a \left(\sum_{i=1}^3 f_a(x_i) b_{2,i} \right), \end{aligned} \quad (18)$$

where $f_a(z)$ is the activation function (let it be a very simple one in the present case: $f_a(z) = z$) of the neurons and $b_{j,i}$ are the weights connecting the i th input neuron to the j th output neuron. Thus the output of the system is

$$\mathbf{y}_1 = f_1(\mathbf{x}_1) = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \quad (19)$$

System 4. The input vector of System 4 is $\mathbf{x}_2 = [x_2 \ x_3] \in \Omega_2 = \omega_2 \times \omega_3$, where $\mathbf{x}_2 \subset \mathbf{x}$ and $\Omega_2 \subset \Omega$.

This system is given by formulas such as

$$Y_2 = f_2(\mathbf{x}_2) = \begin{bmatrix} x_2^2 & 2x_2 \\ 3x_3^{-1} & 5 \end{bmatrix}. \quad (20)$$

System 5. The input vector of System 5 is $\mathbf{x}_3 = x_1 \in \Omega_3 = \omega_1$, where $\mathbf{x}_3 \sqsubset \mathbf{x}$ and $\Omega_3 \sqsubset \Omega$.

This is given by a fuzzy logic model. Assume that two rules are given ($i = 1, 2$):

IF A_i THEN B_i .

Further assume that the membership functions are in Ruspini partition:

$$\begin{aligned} \mu_{A_1}(x_3) &= x_3; \\ \mu_{A_2}(x) &= 1 - x_3; \end{aligned} \quad (21)$$

and the consequent sets are singleton sets located at elements 5 and 6 of the output universe. It is a TS fuzzy model and, therefore, the transfer function (product-sum-gravity) of the model is

$$y_3 = f_3(\mathbf{x}_3) = 5x_3 + 6(1 - x_3). \quad (22)$$

System 6. The input vector of System 6 is $\mathbf{x}_4 = [x_1 \ x_3] \in \Omega_4 = \omega_1 \times \omega_3$, where $\mathbf{x}_4 \sqsubset \mathbf{x}$ and $\Omega_4 \sqsubset \Omega$.

This is a black-box model that can provide y_4 for any input x_1, x_3 .

$$y_4 = f_4(\mathbf{x}_4). \quad (23)$$

(In order to follow all computational steps of the example, let us reveal what is the output of the black-box $y_4 = 3x_1 + x_3$.)

5.2. Conditions of the TP Model Transformation. The goal of the example is to transform all the four systems to TS fuzzy representations (or TP model if the resulting weighting functions cannot be represented as antecedent fuzzy states), with the following conditions:

- (i) All systems must have the same antecedent function system on the input interval of x_1 . The antecedent functions must be in Ruspini partition, namely, in SN and NN type. In order to have a complexity minimised representation, a further requirement is that the number of antecedent functions must be minimal.
- (ii) The same antecedent function system of variable x_2 is predefined for all systems:

$$\mathbf{w}_2^p(x_2) = [w_{2,1}(x_2) \ w_{2,2}(x_2) \ w_{2,3}(x_2)], \quad (24)$$

where “ p ” denotes “predefined,” and

$$\begin{aligned} w_{2,1}(x_2) &= 0.5x_2^2; \\ w_{2,2}(x_2) &= 0.5(1 - x_2)^2 \\ w_{2,3}(x_2) &= 1 - (w_{2,1}(x_2) + w_{2,2}(x_2)). \end{aligned} \quad (25)$$

- (iii) The only requirement for the weighting function system of the input x_3 of each system is that they must be the singular functions of the HOSVD canonical form (orthonormed system ordered by the higher-order singular values). These functions are not representable as antecedent functions of fuzzy sets, since they may take negative values as well. Obviously they will not be the same for all systems.

5.3. Execution of the Proposed TP Model Transformation. It is worth emphasizing again that the previous methods for TP model representation cannot be applied in the present case, since the elements of the input vectors are different.

Step 1.

- (i) Let us define grid G to $\Omega = [0, 1] \times [0, 1] \times [0, 1]$:

$$\mathbf{g}_1 = \mathbf{g}_2 = \mathbf{g}_3 = [0 \ 0.01 \ 0.02 \ \dots \ 1]. \quad (26)$$

Thus the number of points on the discretisation grid is $M_1 = M_2 = M_3 = 101$.

- (ii) Let us discretise the systems over the rectangular grid defined by vectors $\mathbf{g}_n, n = 1, 2, 3$. The discretisation of System l results in $\mathcal{F}_l^{D(\Omega_l, G_l)}, l \in L$. In case of System 3,

$$\mathcal{F}_1^{D(\Omega_1, G_1)} \in \mathbb{R}^{M_1 \times M_2 \times M_3 \times 2}, \quad (27)$$

where the first three dimensions are assigned to the input variables and the last dimension is assigned to the output vector. The discretisation of System 4 yields

$$\mathcal{F}_2^{D(\Omega_2, G_2)} \in \mathbb{R}^{M_2 \times M_3 \times 2 \times 2}, \quad (28)$$

where the first two dimensions are assigned to the input variables x_2, x_3 and the last two dimensions are assigned to the output matrix. The discretisation of System 5 yields the following vector:

$$\mathcal{F}_3^{D(\Omega_3, G_3)} \in \mathbb{R}^{M_3}. \quad (29)$$

The discretisation of System 6 results in

$$\mathcal{F}_4^{D(\Omega_4, G_4)} \in \mathbb{R}^{M_1 \times M_3}, \quad (30)$$

where the first two dimensions are assigned to the input variables x_1, x_3 and the last dimension is assigned to the output vector.

Let us discretise the predefined weighting function as well:

$$\begin{aligned} \mathcal{W}_2^{D(\omega_2, \mathbf{g}_2)} \\ = \begin{bmatrix} w_{2,1}(g_{2,1}) & w_{2,2}(g_{2,1}) & w_{2,3}(g_{2,1}) \\ \vdots & \vdots & \vdots \\ w_{2,1}(g_{2,M_2}) & w_{2,2}(g_{2,M_2}) & w_{2,3}(g_{2,M_2}) \end{bmatrix}. \end{aligned} \quad (31)$$

Step 2.

(i) Dimension x_1

Lay out tensors $\mathcal{F}_l^{D(\Omega_l, G_l)}$, $l \in \{1, 3, 4\}$ in the dimension assigned x_1 :

$$\mathbf{H}_l = \left(\mathcal{F}_l^{D(\Omega_l, G_l)} \right)_{(1)}. \quad (32)$$

Create

$$\mathbf{K} = [\mathbf{H}_1 \ \mathbf{H}_3 \ \mathbf{H}_4]. \quad (33)$$

Execute SVD on \mathbf{K} incorporating SN and NN condition [4] (only nonzero singular values are kept):

$$\mathbf{K} = \mathbf{U}_1 \mathbf{D}' \mathbf{V}'^T. \quad (34)$$

The result of this step to be used later is \mathbf{U}_1 .

(ii) Dimension x_2

Let

$$\mathbf{U}_2 = \mathcal{W}_2^{D(\omega_2, \mathbf{g}_2)}. \quad (35)$$

(iii) Dimension x_3

Lay out tensors $\mathcal{F}_l^{D(\Omega_l, G_l)}$, $l \in \{1, 2, 4\}$ in the dimension assigned x_3 :

$$\mathbf{H}_l = \left(\mathcal{F}_l^{D(\Omega_l, G_l)} \right)_{(3)}. \quad (36)$$

Then execute HOSVD on each \mathbf{H}_l , $l \in \{1, 2, 4\}$ (only the nonzero singular values are kept):

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{U}_{3,1} \mathbf{D}_1 \mathbf{V}_1^T \\ \mathbf{H}_2 &= \mathbf{U}_{3,2} \mathbf{D}_2 \mathbf{V}_2^T \\ \mathbf{H}_4 &= \mathbf{U}_{3,4} \mathbf{D}_4 \mathbf{V}_4^T \end{aligned} \quad (37)$$

The result of this step is $\mathbf{U}_{3,l}$.

(iv) Reconstructing the core tensors

$$\forall l : \mathcal{S}_l = \mathcal{F}_l^{D(\Omega_l, G_l)} \boxtimes_{\Omega_l} (\mathbf{U}_n)^+. \quad (38)$$

Step 3. Let

$$\begin{aligned} \mathcal{W}_1^{D(\omega_1, \mathbf{g}_1)} &= \mathbf{U}_1, \\ \mathcal{W}_{3,l}^{D(\omega_3, \mathbf{g}_3)} &= \mathbf{U}_{3,l}, \end{aligned} \quad (39)$$

$l : \{1, 2, 4\}$.

Then having the discretised tensors and weighting functions of all systems we can numerically reconstruct the weighting functions [4, 5] as

$$\begin{aligned} \mathcal{W}_1^{D(\omega_1, \mathbf{g}_1)} &= \mathbf{U}_1 \longrightarrow \mathbf{w}_1(x_1), \\ \mathcal{W}_{3,l}^{D(\omega_3, \mathbf{g}_3)} &= \mathbf{U}_{3,l} \longrightarrow \mathbf{w}_{3,l}(x_3), \end{aligned} \quad (40)$$

$l : \{1, 2, 4\}$.

Thus, we have achieved our goal:

$$\begin{aligned} f_1(x_1, x_2, x_3) &= \mathcal{S}_1 \times_1 \mathbf{w}_1(x_1) \times_2 \mathbf{w}_2^p(x_2) \times_3 \mathbf{w}_{1,3}(x_3); \\ f_2(x_2, x_3) &= \mathcal{S}_2 \times_2 \mathbf{w}_2^p(x_2) \times_3 \mathbf{w}_{2,3}(x_3); \\ f_3(x_1) &= \mathcal{S}_3 \times_1 \mathbf{w}_1(x_1); \\ f_4(x_1, x_3) &= \mathcal{S}_4 \times_1 \mathbf{w}_1(x_1) \times_3 \mathbf{w}_{4,3}(x_3). \end{aligned} \quad (41)$$

6. Conclusion

The proposed TP model transformation can be executed on a set of models where the dimensionality of the inputs may differ. The proposed TP model transformation has all the advantages of the previous ones, including easy convex hull manipulation, complexity trade-offs, pseudo TP model transformation, and automatic and numerical execution.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the FIEK Program (Center for Cooperation between Higher Education and the Industries at the Széchenyi István University, GINOP-2.3.4-15-2016-00003).

References

- [1] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM Review*, vol. 35, no. 4, pp. 551–566, 1993.
- [2] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [3] P. Baranyi, "TP model transformation as a way to LMI-based controller design," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 2, pp. 387–400, 2004.
- [4] P. Baranyi, Y. Yam, and P. Várlaki, *Tensor Product Model Transformation in Polytopic Model Based Control*, Automation and Control Engineering, CRC Press, Taylor & Francis Group, March 2017.
- [5] P. Baranyi, *TP-Model Transformation-Based-Control Design Frameworks*, Control Engineering, Springer International Publishing, Chengdo, China, 2016.
- [6] P. Baranyi, L. Szeidl, P. Várlaki, and Y. Yam, "Definition of the HOSVD based canonical form of polytopic dynamic models," in *Proceedings of the IEEE International Conference on Mechatronics, ICM '06*, pp. 660–665, Hungary, July 2006.
- [7] P. Baranyi, L. Szeidl, and P. Várlaki, "Numerical reconstruction of the HOSVD based canonical form of polytopic dynamic models," in *Proceedings of the 10th International Conference on Intelligent Engineering Systems '06*, pp. 196–201, Agadir, Morocco, June 2006.
- [8] P. Baranyi, "The generalized TP model transformation for T–S fuzzy model manipulation and generalized stability verification," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 4, pp. 934–948, 2014.

- [9] D. Tikk, P. Baranyi, R. Patton, and J. Tar, "Approximation capability of TP model forms," *Australian Journal of Intelligent Information Processing Systems*, vol. 8, no. 3, pp. 155–163, 2004.
- [10] P. Baranyi and Y. Yam, "Fuzzy Rule Base Reduction," in *Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications*, D. Ruan and E. E. Kerre, Eds., ch. 7, pp. 135–160, Kluwer, 2000.
- [11] Y. Yam, P. Baranyi, and C.-T. Yang, "Reduction of fuzzy rule base via singular value decomposition," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 120–132, 1999.
- [12] K. Tanaka, T. Ikeda, and H. O. Wang, "Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 2, pp. 250–265, 1998.
- [13] K. Tanaka and H. Wang, "Fuzzy regulators and fuzzy observers: a linear matrix inequality approach," in *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 1315–1320, San Diego, CA, USA, 2001.
- [14] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets and Systems*, vol. 45, no. 2, pp. 135–156, 1992.
- [15] C. Scherer and S. Weiland, "Linear Matrix Inequalities in Control," in *Lecture Notes*, Dutch Institute for Systems and Control, Delft, The Netherlands, 2000.
- [16] J. Cui, K. Zhang, and T. Ma, "An efficient algorithm for the tensor product model transformation," *International Journal of Control, Automation, and Systems*, vol. 14, no. 5, pp. 1205–1212, 2016.
- [17] S. Nagy, Z. Petres, P. Baranyi, and H. Hashimoto, "Computational relaxed TP model transformation by restriction of the computation to subspaces of the dynamic model," in *Proceedings of the ISCIII'07: 3rd International Symposium on Computational Intelligence and Intelligent Informatics*, pp. 99–104, mar, March 2007.
- [18] A. Szollosi and P. Baranyi, "Influence of the tensor product model representation of qLPV models on the feasibility of linear matrix inequality," *Asian Journal of Control*, vol. 18, no. 4, pp. 1328–1342, 2016.
- [19] A. Szollosi and P. Baranyi, "Improved control performance of the 3-DOF aeroelastic wing section: a TP model based 2D parametric control performance optimization," *Asian Journal of Control*, vol. 19, no. 2, pp. 450–466, 2017.
- [20] A. Szollosi and P. Baranyi, "Influence of the Tensor Product Model Representation of qLPV Models on the Feasibility of Linear Matrix Inequality Based Stability Analysis," *Asian Journal of Control*, 2017.
- [21] X. Liu, Y. Yu, Z. Li, H. H. C. Iu, and T. Fernando, "An Efficient Algorithm for Optimally Reshaping the TP Model Transformation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 10, pp. 1187–1191, 2017.
- [22] J. Kuti, P. Galambos, and P. Baranyi, "Minimal volume simplex (MVS) polytopic model generation and manipulation methodology for TP model transformation," *Asian Journal of Control*, vol. 19, no. 1, pp. 289–301, 2017.
- [23] P. Várkonyi, D. Tikk, P. Korondi, and P. Baranyi, "A new algorithm for RNO-INO type Tensor Product model representation," in *Proceedings of the INES'05: IEEE 9th International Conference on Intelligent Engineering Systems*, pp. 263–266, September 2005.
- [24] X. Liu, X. Xin, Z. Li, and Z. Chen, "Near Optimal Control Based on the Tensor-Product Technique," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 5, pp. 560–564, 2017.
- [25] X. Liu, Y. Yu, Z. Li, H. H. C. Iu, and T. Fernando, "A novel constant gain Kalman filter design for nonlinear systems," *Signal Processing*, vol. 135, pp. 158–167, 2017.
- [26] X. Liu, Y. Yu, Z. Li, and H. H. C. Lu, "Polytopic h1 filter design and relaxation for nonlinear systems via tensor product technique," in *Signal Processing*, vol. 127, pp. 191–205, 2016.
- [27] P. S. Saikrishna, R. Pasumathy, and N. P. Bhatt, "Identification and Multivariable Gain-Scheduling Control for Cloud Computing Systems," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 792–807, 2017.
- [28] Q. Zhang and X. Zhang, *Vapor Compression Cycle Control for Automotive Air Conditioning Systems with a Linear Parameter Varying Approach*, Cornell University Library, 2017.
- [29] G. Zhao, D. Wang, and Z. Song, "A novel tensor product model transformation-based adaptive variable universe of discourse controller," *Journal of The Franklin Institute*, vol. 353, no. 17, pp. 4471–4499, 2016.
- [30] W. Qin, B. He, Q. Qin, and G. Liu, "Robust active controller of hypersonic vehicles in the presence of actuator constraints and input delays," in *Proceedings of the 35th Chinese Control Conference, CCC 2016*, pp. 10718–10723, Chengdu, China, July 2016.
- [31] Q. Weiwei, H. Bing, L. Gang, and Z. Pengtao, "Robust model predictive tracking control of hypersonic vehicles in the presence of actuator constraints and input delays," *Journal of The Franklin Institute*, vol. 353, no. 17, pp. 4351–4367, 2016.
- [32] T. Wang and B. Liu, "Different polytopic decomposition for visual servoing system with LMI-based Predictive Control," in *Proceedings of the 35th Chinese Control Conference, CCC 2016*, pp. 10320–10324, China, July 2016.
- [33] T. Wang and W. Zhang, "The visual-based robust model predictive control for two-DOF video tracking system," in *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, pp. 3743–3747, China, May 2016.
- [34] T. Jiang and D. Lin, "Tensor product model-based gain scheduling of a missile autopilot," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 59, no. 3, pp. 142–149, 2016.
- [35] A. Hajiloo, *Robust and multi-objective model predictive control design for nonlinear systems [Ph.D. thesis]*, Concordia University, 2016.
- [36] A. Hajiloo and W. F. Xie, "The stochastic robust model predictive control of shimmy vibration in aircraft landing gears," *Asian Journal of Control*, vol. 17, no. 2, pp. 476–485, 2015.
- [37] Y. Yu, Z. Li, J. Li, L. Li, and X. Liu, "H ∞ polytopic filter design for nonlinear systems," in *Proceedings of the 35th Chinese Control Conference (CCC '16)*, pp. 620–625, Chengdu, China, July 2016.
- [38] P. Bo, Y. Lingyu, and Y. Xiaoke, "LPV-MRAC Method for Aircraft with Structural Damage," in *Proceedings of the 35th Chinese Control Conference (CCC)*, pp. 3262–3267, Chengdu, China, July 2016.
- [39] V. C. S. Campos, F. O. Souza, L. A. B. Torres, and R. M. Palhares, "New stability conditions based on piecewise fuzzy lyapunov functions and tensor product transformations," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 748–760, 2013.
- [40] S. Kuntanapreeda, "Tensor product model transformation based control and synchronization of a class of fractional-order chaotic systems," *Asian Journal of Control*, vol. 17, no. 2, pp. 371–380, 2015.

- [41] R.-E. Precup, E. M. Petriu, M.-B. Radac, S. Preitl, L.-O. Fedorovici, and C.-A. Dragos, "Cascade control system-based cost effective combination of tensor product model transformation and fuzzy control," *Asian Journal of Control*, vol. 17, no. 2, pp. 381–391, 2015.
- [42] G. Zhao, H. Li, and Z. Song, "Tensor product model transformation based decoupled terminal sliding mode control," *International Journal of Systems Science*, vol. 47, no. 8, pp. 1791–1803, 2016.
- [43] P. Korondi, "Tensor product model transformation-based sliding surface design," *Acta Polytechnica Hungarica*, vol. 3, no. 4, pp. 23–35, 2006.
- [44] G. Eigner, I. Bőjthe, P. Pausits, and L. Kovács, "Investigation of the TP modeling possibilities of the Hovorka TIDM model," in *Proceedings of the 15th IEEE International Symposium on Applied Machine Intelligence and Informatics, SAMI 2017*, pp. 259–264, Slovakia, January 2017.
- [45] G. Eigner, P. Pausits, and L. Kovacs, "Control of TIDM via tensor product-based framework," in *Proceedings of the 17th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2016*, pp. 55–60, Hungary, November 2016.
- [46] G. Eigner, I. Rudas, A. Szakal, and L. Kovacs, "Tensor product based modeling of tumor growth," in *Proceedings of the 2017 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 900–905, Banff, AB, October 2017.
- [47] L. Kovacs and G. Eigner, Usability of the tensor product based modeling in the modeling of diabetes mellitus, manuscript in preparation, 2016.
- [48] G. Eigner, I. J. Rudas, and L. Kovacs, "Investigation of the TP-based modeling possibility of a nonlinear ICU diabetes model," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pp. 3405–3410, Hungary, October 2016.
- [49] L. Kovacs and G. Eigner, "Convex polytopic modeling of diabetes mellitus: A Tensor Product based approach," in *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pp. 3393–3398, Hungary, October 2016.
- [50] T. T. Wang, W. F. Xie, G. D. Liu, and Y. M. Zhao, "Quasi-min-max model predictive control for image-based visual servoing with tensor product model transformation," *Asian Journal of Control*, vol. 17, no. 2, pp. 402–416, 2015.
- [51] J. Kuti, P. Galambos, and Á. Miklós, "Output feedback control of a dual-excenter vibration actuator via qLPV model and TP model transformation," *Asian Journal of Control*, vol. 17, no. 2, pp. 432–442, 2015.
- [52] J. Pan and L. Lu, "TP model transformation via sequentially truncated higher-order singular value decomposition," *Asian Journal of Control*, vol. 17, no. 2, pp. 467–475, 2015.
- [53] J. Matuško, Š. Ileš, F. Kolonić, and V. Leši, "Control of 3D tower crane based on tensor product model transformation with neural friction compensation," *Asian Journal of Control*, vol. 17, no. 2, pp. 443–458, 2015.
- [54] V. c. Campos, L. A. Tórres, and R. M. Palhares, "Revisiting the TP model transformation: interpolation and rule reduction," *Asian Journal of Control*, vol. 17, no. 2, pp. 392–401, 2015.
- [55] G. Zhao, K. Sun, and H. Li, "Tensor product model transformation based adaptive integral-sliding mode controller: Equivalent control method," *The Scientific World Journal*, vol. 2013, Article ID 726963, 2013.
- [56] S. Chumalee and J. F. Whidborne, "Gain-Scheduled H ∞ Control for Tensor Product Type Polytopic Plants," *Asian Journal of Control*, vol. 17, no. 2, pp. 417–431, 2015.
- [57] G. Zhao, S. Huang, Y. Zhang, T. Zhang, and Y. Zhang, "Tensor product model transformation based fractional decoupled sliding-mode control for cart-pole system with time-varying sliding surfaces and Dahl friction model," in *Proceedings of the 29th Chinese Control and Decision Conference, CCDC 2017*, pp. 3582–3589, China, May 2017.
- [58] J. Klespitz, I. J. Rudas, and L. Kovacs, "LMI-Based Feedback Regulator Design via TP Transformation for Fluid Volume Control in Blood Purification Therapies," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 2615–2619, Hong Kong, October 2015.
- [59] G. Zhao, C. Zhao, and D. Wang, "Tensor product model transformation based integral sliding mode control with reinforcement learning strategy," in *Proceedings of the 33rd Chinese Control Conference, CCC 2014*, pp. 77–82, China, July 2014.
- [60] S. Huang, G. Zhao, Y. Yuan, Q. Ren, and Y. Liu, "Intelligent tensor product mode transformation-based three-sliding-surface sliding mode controller design," in *Proceedings of the 34th Chinese Control Conference, CCC 2015*, pp. 3258–3263, China, July 2015.
- [61] Á. Takács, T. Haidegger, P. Galambos, J. Kuti, and I. J. Rudas, "Nonlinear soft tissue mechanics based on polytopic Tensor Product modeling," in *Proceedings of the IEEE 14th International Symposium on Applied Machine Intelligence and Informatics, SAMI '16*, pp. 211–215, Slovakia, January 2016.
- [62] B. Takarics and Y. Yam, "Robust Grid Point-Based Control Design for LPV Systems via Unified TP Transformation," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC '15*, pp. 2626–2631, Hong Kong, October 2015.
- [63] Z. He, M. Yin, and Y.-P. Lu, "Tensor product model-based control of morphing aircraft in transition process," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, no. 2, pp. 378–391, 2016.
- [64] J. Chen, R. Li, and C. Cao, "Convex polytopic modeling for flexible joints industrial robot using TP-model transformation," in *Proceedings of the IEEE International Conference on Information and Automation, ICIA '14*, pp. 1046–1050, Hailar, China, July 2014.
- [65] R.-E. Precup, C.-A. Dragos, S. Preitl, M.-B. Radac, and E. M. Petriu, "Novel tensor product models for automatic transmission system control," *IEEE Systems Journal*, vol. 6, no. 3, pp. 488–498, 2012.
- [66] K. K. Wu and Y. Yam, "Control stability of TP model transformation design via probabilistic error bound of plant model," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC '13*, pp. 1259–1264, UK, October 2013.
- [67] R.-E. Precup, L.-T. Dioanca, E. M. Petriu, M.-B. Rădac, S. Preitl, and C.-A. Dragos, "Tensor Product-based real-time control of the liquid levels in a three tank system," in *Proceedings of the 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2010*, pp. 768–773, Canada, July 2010.
- [68] K. Széll, A. Czmerk, and P. Korondi, "Automatika : časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije," *Identifikacija modela trenja s histerezom korištenjem tenzorskog produkta*, vol. 55, no. 4, pp. 463–473, 2015.

- [69] F. Kolonić, A. Poljugan, and I. Petrović, "Tensor product model transformation-based controller design for gantry crane control system - An application approach," *Acta Polytechnica Hungarica*, vol. 3, no. 4, pp. 95–112, 2006.
- [70] F. Kolonie and A. Poljugan, "Experimental control design by TP model transformation," in *Proceedings of the 2006 IEEE International Conference on Mechatronics, ICM*, pp. 666–671, Hungary, July 2006.
- [71] P. Grof and Y. Yam, "Furuta Pendulum-A Tensor Product Model-Based Design Approach Case Study," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pp. 2620–2625, Hong Kong, October 2015.
- [72] Z. Petres, Polytopic decomposition of linear parameter-varying models by tensor-product model transformation, 2006.
- [73] P. Galambos, *TP Model Transformation Based Control Design for Time-Delay Systems: Application in Telemanipulation*, 2013.
- [74] C. Ariño and A. Sala, "Relaxed LMI conditions for closed-loop fuzzy systems with tensor-product structure," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1036–1046, 2007.
- [75] P. Gróf, B. Takarics, Z. Petres, and P. Korondi, "Tensor product model transformation based control of a pneumatic cylinder," in *Proceedings of the 8th International Symposium on Applied Machine Intelligence and Informatics (SAMi)*, vol. 2008.
- [76] F. Yang, Z. Chen, X. Liu, and B. Liu, "A new constant gain Kalman filter based on TP model transformation," in *Proceedings of the Chinese Intelligent Automation Conference*, vol. 254, pp. 305–312, Springer, 2013.
- [77] Š. Ileš, J. Matuško, and F. Kolonić, "Tensor product transformation based speed control of permanent magnet synchronous motor drives," in *17th international conference on electrical drives and power electronics, EDPE '11 (5th Joint Slovak-Croatian Conference)*, 2011.
- [78] S. Ileš, J. Matuško, and F. Kolonić, "TP transformation based control of rotary pendulum," in *Proceedings of the 34th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO '11*, pp. 833–839, IEEE, Opatija, Croatia, May 2011.
- [79] R. Precup, C. Dragos, S. Preitl, M. Radac, and E. Petriu, "Tensor product models for automotive applications," in *Proceedings of the 14th International Conference on System Theory and Control (ICSTC '10)*, 2010.
- [80] P. Korondi, "Sector sliding mode design based on tensor product model transformation," in *Proceedings of the INES 2007 - 11th International Conference on Intelligent Engineering Systems*, pp. 253–258, Hungary, July 2007.
- [81] Y. Kunii, B. Solvang, G. Sziebig, and P. Korondi, "Tensor product transformation based friction model," in *Proceedings of the INES 2007 - 11th International Conference on Intelligent Engineering Systems*, pp. 259–264, Hungary, July 2007.
- [82] P. Korondi, "Sliding sector design based on tensor product model transformation," *Transaction on Automatic Control and Computer Science*, vol. 51, no. 1, pp. 101–108, 2006.
- [83] P. Korondi, P. Bartal, and F. Koloni, "Friction model based on tensor product transformation," in *Proceedings of the 7th International Symposium of Hungarian Researchers on Computational Intelligence*, 2006.
- [84] G. Hancke and A. Szeghegyi, "nonlinear control via tp model transformation: The tora system example," in *Proceedings of the Symposium on Applied Machine Intelligence (SAMi, '04)*, pp. 333–340, Herlany, Slovakia, 2004.
- [85] G. Hancke and A. Szeghegyi, "Application study of the tp model transformation in the control of an inverted pendulum," in *Proceedings of the in International Conference on Computational Cybernetics (ICCC)*, 2003.
- [86] S. Ileš, F. Kolonić, and J. Matuško, "Linear matrix inequalities based H_∞ control of gantry crane using tensor product transformation," in *Proceedings of the 18th International Conference on Process Control*, pp. 92–99, Tatranská Lomnica, Slovakia, 2011.
- [87] A. Rövid, L. Szeidl, and P. Várlaki, "On tensor-product model based representation of neural networks," in *Proceedings of the 15th International Conference on Intelligent Engineering Systems, INES 2011*, pp. 69–72, Slovakia, June 2011.
- [88] K. Tanaka, H. Yoshida, H. Ohtake, and H. O. Wang, "A sum-of-squares approach to modeling and control of nonlinear dynamical systems with polynomial fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 4, pp. 911–922, 2009.
- [89] K. Tanaka, H. Ohtake, T. Seo, M. Tanaka, and H. O. Wang, "Polynomial fuzzy observer designs: A sum-of-squares approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 5, pp. 1330–1342, 2012.

Research Article

The Spiral Discovery Network as an Automated General-Purpose Optimization Tool

Adam B. Csapo 

Department of Informatics, Széchenyi István University, Győr, Hungary

Correspondence should be addressed to Adam B. Csapo; csapo.adam@sze.hu

Received 29 September 2017; Accepted 22 January 2018; Published 12 March 2018

Academic Editor: Kevin Wong

Copyright © 2018 Adam B. Csapo. 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 Spiral Discovery Method (SDM) was originally proposed as a cognitive artifact for dealing with black-box models that are dependent on multiple inputs with nonlinear and/or multiplicative interaction effects. Besides directly helping to identify functional patterns in such systems, SDM also simplifies their control through its characteristic spiral structure. In this paper, a neural network-based formulation of SDM is proposed together with a set of automatic update rules that makes it suitable for both semiautomated and automated forms of optimization. The behavior of the generalized SDM model, referred to as the Spiral Discovery Network (SDN), and its applicability to nondifferentiable nonconvex optimization problems are elucidated through simulation. Based on the simulation, the case is made that its applicability would be worth investigating in all areas where the default approach of gradient-based backpropagation is used today.

1. Introduction

The question of how to gain an understanding of the operation of a system arises naturally in a wide range of application areas. However, this question is not always easy to answer, in part because different use cases favor different approaches. While a set of closed formulae might be useful when it comes to predicting exactly how the system will operate under specific conditions, they may be difficult to formulate when the conditions themselves and/or their effects are hard to characterize. In such cases, black-box identification and heuristic modelling approaches are often used.

The neural network presented in this paper, referred to as the Spiral Discovery Network (SDN), is a generalized version of the Spiral Discovery Method, which is a semiautomated cognitive artifact [1, 2]. SDM originally served the purpose of helping users to discover systematic relationships between multiple inputs to a system and the system's output behavior, even when the inputs have nonlinear effects and multiplicative cross-effects on the output. The goal in extending the SDM model is to extend its applicability to automated settings in which neural networks (or other parametric black-box models) tune their behavior based on a set of functional

constraints, such as requirements on the structure of their output or other external error feedback signals.

Through the formulation proposed in this paper, it turns out that SDM is applicable whenever a data-driven approach is available to the identification of a system and whenever the effects of various changes in its inputs can be evaluated in a reasonable amount of time. When the evaluations are performed by humans, SDM shows motivations and characteristics similar to those of the paradigm of interactive evolutionary computation [3, 4]; however, it shows differences in terms of the logic through which it helps to discover parametric spaces. Its extended version, SDN, is also more generally applicable by allowing for automated evaluations. As discussed in the conclusions of the paper, SDN is noteworthy in that it does not rely on gradient information, a feature that can be seen to reduce the complexity of the required computations, as well as being potentially helpful in cases where the performance of gradient-based solutions is far from optimal (for a detailed discussion on such cases, the reader is referred to [5]).

The paper is structured as follows. Section 2 provides a short overview of the literature on nonconvex optimization in order to position the relevance of this work with respect to

earlier results. Section 3 then briefly reviews the background of the original Spiral Discovery Method (SDM). Section 4 introduces the tensor-algebra based numerical structures behind the original SDM formulation. In Section 5, the neural network-based Spiral Discovery Network (SDN) is introduced. A simulation example is provided in Section 6 in order to demonstrate the viability of the model in handling nonconvex and nondifferentiable optimization problems. Finally, Section 7 concludes the paper.

2. Historical Overview

Nonconvex optimization is a broad field of mathematics that finds many applications in engineering tasks where the goal is to find sufficiently good solutions on high-dimensional parametric manifolds. One of the most relevant examples today is finding useful architectures for (deep) neural networks or other kinds of graphical models, as well as finding the right set of parameters with which to operate them. The common approach in solving such problems is to iteratively refine a candidate solution in a way that incrementally improves upon it in terms of a globally defined loss function: this is known as gradient descent [6].

The general idea of gradient descent can be highly successful on parametric landscapes that are associated with a clearly defined cost function and contain no more than a small number of local minima in terms of that function. However, as soon as the value of a cost function becomes difficult to interpret or the cost function becomes so intractable that it is computationally difficult to determine its gradients and/or it produces an intractably large number of local minima, the naive solution of gradient-based iterative optimization often starts to break down.

The problem of dealing with local minima can be addressed to some degree by finding good trade-offs between exploration and exploitation, that is, by modifying the gradient descent approach slightly to counteract situations where the optimization process might slow down or stop. This approach is reflected in a host of existing solutions. One fruitful idea was to experiment with the scaling factor of the gradient, for example, by making it adaptive to changes in sign via the concept of “momentum” [7–9] or by making it specific to the different dimensions in the parameter space [10, 11]. Other ideas include the normalization of inputs across layers and batches (specifically in training neural network models) [12] or by simply adding noise to the gradients [13].

The above solutions notwithstanding, the general idea of modifying a candidate solution in the direction of the negative gradient of a loss function has largely remained unchallenged. Only recently have the remarks of G. Hinton and other highly regarded researchers become widely publicized, which suggest that gradient descent, at least based on backpropagation, may prove not to be the ultimate solution for training neural networks (see, e.g., the article entitled “Why We Should Be Deeply Suspicious of BackPropagation” by C. E. Perez on <https://medium.com/intuitionmachine/the-deeply-suspicious-nature-of-backpropagation-9bed5e2b085e>).

In this paper, the earlier idea of the Spiral Discovery Method is extended to the domain of automatic training in neural networks through a neural architecture. Instead of relying on gradients to update its search location, the method follows a hierarchical hyperspiral structure within the parametric space, thus gaining insight into search directions that may be fruitful.

3. Original Problem Formulation Behind SDM

In this section, we consider a generic formulation of the class of problems to which the original Spiral Discovery Method (SDM) can be applied. To this end, we will make use of the following concepts and notations:

- (i) A vector of *generation parameters* $\mathbf{g} \in \mathbb{R}^G$
- (ii) A perceptually accessible *output* $o \in \mathbb{R}$
- (iii) A *system transfer function* $S : \mathbb{R}^G \rightarrow \mathbb{R}$, which evaluates generation parameter vectors to produce perceptually accessible outputs
- (iv) An *evaluation function* $E : \mathbb{R} \rightarrow \mathbb{R}$, which associates perceptually accessible outputs with a real number referred to as the *perceptual value* of a given output
- (v) A set $\mathcal{D} = \{(\mathbf{g}_1, \mathbf{v}_1), (\mathbf{g}_M, \mathbf{v}_M)\}$ referred to as the *data set*, which contains tuples of generation parameter vectors and perceptual values.

In the original problem formulation, the goal is to find a set of generation parameter vectors that are suitable for the generation of a controlled set of outputs, controlled, that is, from the perspective of the perceptually driven evaluation function. Most often, the problem would present itself in such a form that a user is given a perceptual value, v' , and the goal is to find a generation parameter vector, \mathbf{g}' , suitable for the generation of an output that yields v' as its perceptual value. In general, solving this problem amounts to more than just inverting the system transfer function (if such an inversion were even possible to begin with), as the relationship between system output and its perception value, which is usually much too complex to be formulated analytically, also must be taken into account.

Application areas in which the above formulation is of interest include the following:

- (i) Tuning a set of parameters to a uni- or multimodal synthesis algorithm for perceptual continuity: for example, in a virtual reality with object-to-sound and object-to-vibration mappings, given a set of parameters used to generate audio signals and vibration patterns for spherical and block-like objects, the goal might be to find an appropriate set of generation parameters for certain kinds of polyhedra, conceptually situated “somewhere between” spheres and blocks.
- (ii) Controlling inputs to complex black-box models based on derived quantifications of success: for example, inputs to a multispeaker system or a distributed

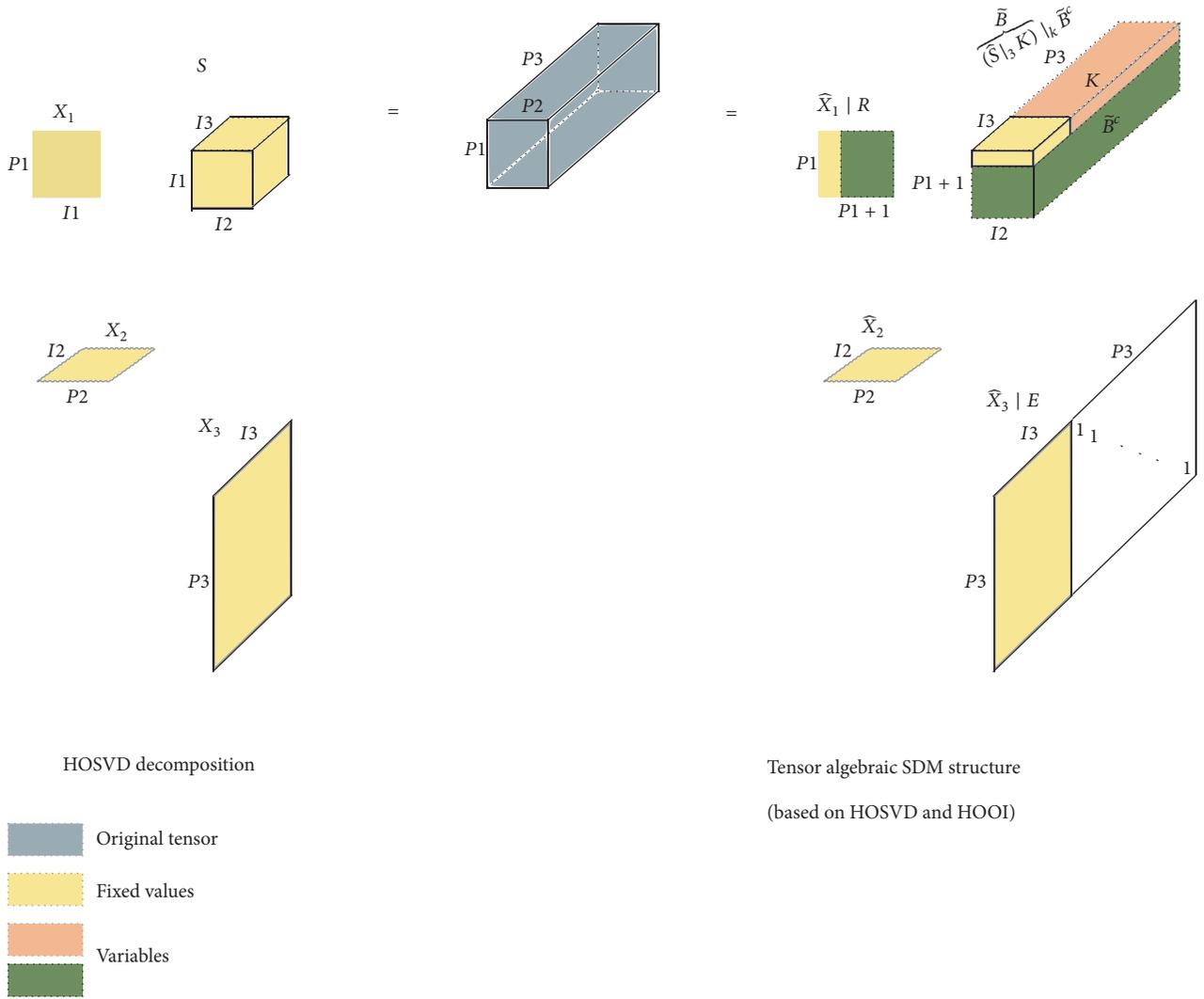


FIGURE 1: Tensor algebraic formulation of the Spiral Discovery Method based on higher-order singular value decomposition and higher-order orthogonal iteration (figure adapted from [2]).

heating system in a large auditorium might be fine-tuned in order to accommodate extrinsic requirements of comfort and cost-effectiveness.

The overall characteristic of the problem formulation is that it encompasses problems where a set of parameters can be used to control a model, usually a black-box model, whose functionality can best be evaluated indirectly through effects that are not well understood, for example, perceptual effects, qualitative measures such as comfort, or aggregated measures such as cost-effectiveness.

It is clear that such formulation can be easily generalized to cases where the evaluation is performed not by humans, but by any kind of automatic process extrinsic to the system. Such processes might still involve a weaker link to human perception or more generally to qualitative cognitive measures but would nevertheless be directly or indirectly measurable and interpretable.

4. Tensor Algebraic Formulation of the Spiral Discovery Method

The original formulation of SDM is in a tensor algebraic form, shown in Figure 1. It is based on the discretization of a hypothetical function that maps vectors of perceptual values \mathbf{v}_i to generation parameters \mathbf{g}_i . In most cases, this function cannot be expressed analytically and might even be different depending on various circumstances, such as the user performing the evaluation. At the same time, a discretized form of the function can often be sampled through experiments (this idea is inspired by the Tensor Product model [14–16]). The discretization is stored in a tensor, \mathcal{F} , such that all dimensions, save for the last one, correspond to discrete gradations along perceptual scales (e.g., “roughness,” “softness,” “degree of comfort,” or “cost-effectiveness”), while the last dimension stores G -dimensional generation parameter vectors corresponding to the perceptual configurations.

The above described tensor, \mathcal{F} , is first decomposed into a core tensor and a set of weighting matrices based on the higher-order singular value decomposition (HOSVD) [17]. This is followed by an iterative rank-reduction step, known as higher-order orthogonal iteration (HOOI) [18], which creates a rank-reduced approximation of the complete system, such that its outputs are controlled by only a single parameter in the perceptual dimension of interest. The twist in the approach is that the “meaning” of this parameter, in other words, the hyperplane along which it influences the system, is cyclically changed through a numerical reconstruction of the system and the systematic manipulation of the core tensor.

The conceptual background of SDM can be well described through a 2-dimensional numerical example. Consider the function described by \mathbf{F} :

$$\mathbf{F} = \begin{pmatrix} 5 & 2 \\ 3 & 3 \\ 10 & 5 \end{pmatrix} \quad (1)$$

in which there are 2 generative parameters for 3 different perceptual gradations. Using singular value decomposition (SVD, instead of HOSVD because we are in case of two dimensions), we obtain

$$\begin{aligned} \mathbf{F} &= \mathbf{S} \times_1 \mathbf{U} \times_2 \mathbf{V} \\ &= \begin{pmatrix} 13.04 & 0 \\ 0 & 1.4 \end{pmatrix} \times_1 \begin{pmatrix} 0.41 & -0.39 \\ 0.31 & 0.91 \\ 0.86 & -0.14 \end{pmatrix} \\ &\quad \times_2 \begin{pmatrix} 0.89 & -0.46 \\ 0.46 & 0.89 \end{pmatrix}. \end{aligned} \quad (2)$$

Optimal rank-reduction in the 2-dimensional case consists simply of removing the second column of \mathbf{S} and the second row of \mathbf{V} or setting $\mathbf{S}(2, 2) = 0$ (thus, in this simple case of two dimensions, HOOI needs not be used). Once $\mathbf{S}(2, 2) = 0$, the second row of the core tensor consists of all zeros and can be removed (as a result, the second column of \mathbf{U} is also removed).

After augmenting the matrix of singular values and the weighting matrices as specified by SDM, we obtain

$$\begin{aligned} \tilde{\mathbf{S}} &= \begin{pmatrix} 13.04 & 0 & a & b \\ d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \end{pmatrix} \\ \tilde{\mathbf{U}} &= \begin{pmatrix} 0.41 & r_{11} & r_{12} & r_{13} \\ 0.31 & r_{21} & r_{22} & r_{23} \\ 0.86 & r_{31} & r_{32} & r_{33} \end{pmatrix} \\ \tilde{\mathbf{V}} &= \begin{pmatrix} 0.89 & -0.46 & 1 & 0 \\ 0.46 & 0.89 & 0 & 1 \end{pmatrix}. \end{aligned} \quad (3)$$

If a and b and the random values in the second, third, and fourth columns of $\tilde{\mathbf{U}}$ are specified, the second, third, and fourth rows of $\tilde{\mathbf{S}}$ can be calculated such that the original system is reconstructed. Then, by modifying just the first column of weighting matrix $\tilde{\mathbf{U}}$, a linear subspace of the original 2-dimensional space can be explored, starting from any of the three perceptual gradations. By separating what is constant from the parts of the equation that are changed, we obtain

$$\tilde{\mathbf{F}} = \begin{bmatrix} \tilde{u}_{11} \\ \tilde{u}_{21} \\ \tilde{u}_{31} \end{bmatrix} [13.04 \ 0 \ a \ b] \tilde{\mathbf{V}}^T + \mathbf{R} \mathbf{D} \tilde{\mathbf{V}}^T. \quad (4)$$

Because the second term is a constant and the first one only depends on the first column of $\tilde{\mathbf{U}}$, the “slope” of the equation, that is, the ratio of change between the second and first output (as the first column of weighting matrix $\tilde{\mathbf{U}}$ is modified), can be written as

$$\text{slope}_{xy} = \frac{(13.04 \ 0 \ a \ b) \tilde{\mathbf{v}}_2^T}{(13.04 \ 0 \ a \ b) \tilde{\mathbf{v}}_1^T} = \frac{13.04 \cdot 0.46 + b}{13.04 \cdot 0.89 + a}. \quad (5)$$

It is clear that based on (5) the slope can be set to any value just by modifying the values of a and b . If the values of a and b are changed systematically between two extreme values, the slope of discovery will also oscillate along the principal component of the original matrix.

5. The Spiral Discovery Network Cell: A Neural Network-Based Formulation of SDM

The key observation of this paper is that SDM can be formulated in much simpler and at the same time more powerful terms using neural networks. The recurrent model shown in Figure 2 is capable of producing systematic, cyclic patterns similar to the original formulation, but at the same time it is adaptive based on a set of external feedback signals. The cell consists of the following modules:

- (i) A *timer* that functions as a modulo counter for updating the state of the cell at discrete time steps
- (ii) A *perturbation module* that determines the direction in which and the extent to which the slope of exploration is to be modified at each time step
- (iii) A *hypervisor module* that refreshes the hyperparameters of the perturbation module based on feedback signals

A graphical representation of an SDN cell and its modules is shown in Figure 2. The updated activation at time t is

$$\mathbf{a}^{(t)} = \alpha^{(t)} \mathbf{x} + \beta \mathbf{p}^{(t)}, \quad (6)$$

and other metaheuristic approaches, such as genetic algorithms [23–25]. PSO and genetic algorithms are somewhat similar to SDN cells in the sense that exploration evolves towards more promising areas of the parametric space. However, the two categories of approaches are also different in the way that they make a compromise between exploration and exploitation: even when evolving towards more promising regions, SDN cells still represent alternative regions to an extent that depends on how varied the obtained feedback values were (exploration); it is the principal direction of the next cycle that in turn influences exploitation.

6. Simulation Example

As a simulation example, we consider a surface described by two parameters, x and y , that can take values of $(0, 10]$. The surface is expressed through the following relationship (see also Figure 3):

$$z = \begin{cases} 500 & \text{if } x, y \notin (0, 10] \times (0, 10] \\ 70 & \text{if } x, y \in [1, 1.5] \times [2.75, 4.5] \\ -10 & \text{if } x, y \in [3.25, 3.5] \times [3.5, 4.25] \\ (x-5)^2 + \dots & \\ -2(y-2) + \dots & \\ x + e^{1/(x+y)} & \text{otherwise.} \end{cases} \quad (9)$$

Figure 4 shows that the minimum location of the search (and parameters thereof) was found as early as in the 7th cycle, without recourse to any kind of gradient information. Although no location for the exact minimum (-10) was found, it can be argued that the obtained results come quite close to achieving this, for two reasons:

- (i) The range of values of the loss function was between 70 and -10 ; hence the value of -8.44 falls within 2% of error.
- (ii) The search itself was unconstrained (i.e., was not guided by the knowledge that only values between 0 and 10 were to be considered on the x - and y -axes): of course, as expected, the fact that locations outside of the specified range had a loss value of 500 helped to guide the search.

Although rudimentary, the example shows the potential value of SDM in dealing with optimization problems that are nonconvex and nondifferentiable.

7. Conclusions

In this paper, an extended, automated variant of the Spiral Discovery Method is proposed. The variant is formulated as a neural network, or rather as a component thereof, and is referred to as the Spiral Discovery Network (SDN) cell. The model of SDN cells incorporates several beneficial properties.

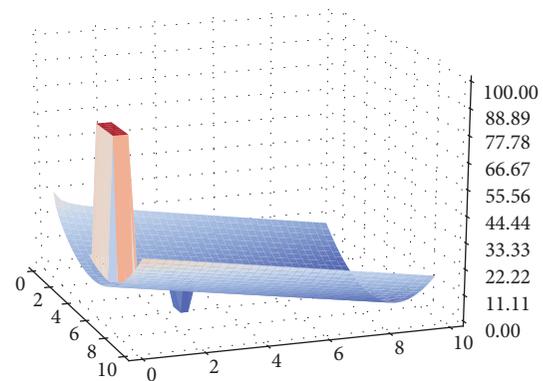
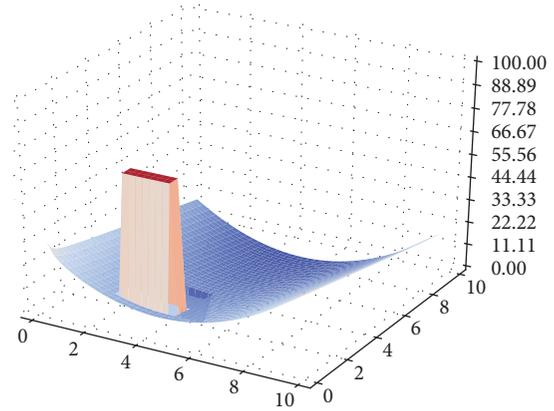


FIGURE 3: Two views of the complex surface to be minimized in simulation example.

First, it is capable of exploring large areas of parametric spaces through a parametric hyperspiral structure, such that the hyperspiral structure itself changes through adaptive cycles. Second, it can rely on any kind of quantitative (perhaps even qualitative) feedback, not only gradient information, to achieve its adaptivity. These properties combined make SDN cells a candidate solution for optimization problems in which the parametric space is nonconvex and potentially even nondifferentiable. A rudimentary simulation was described in the paper to demonstrate the capabilities of SDN cells. One possible avenue of investigation as part of future work would be to consider how SDN cells might be used as part of a network to further improve optimization performance.

Conflicts of Interest

The author declares that they have no conflicts of interest.

Acknowledgments

This work was supported by the FIEK program (Center for Cooperation between Higher Education and the Industries at Széchenyi István University, GINOP-2.3.4-15-2016-00003).

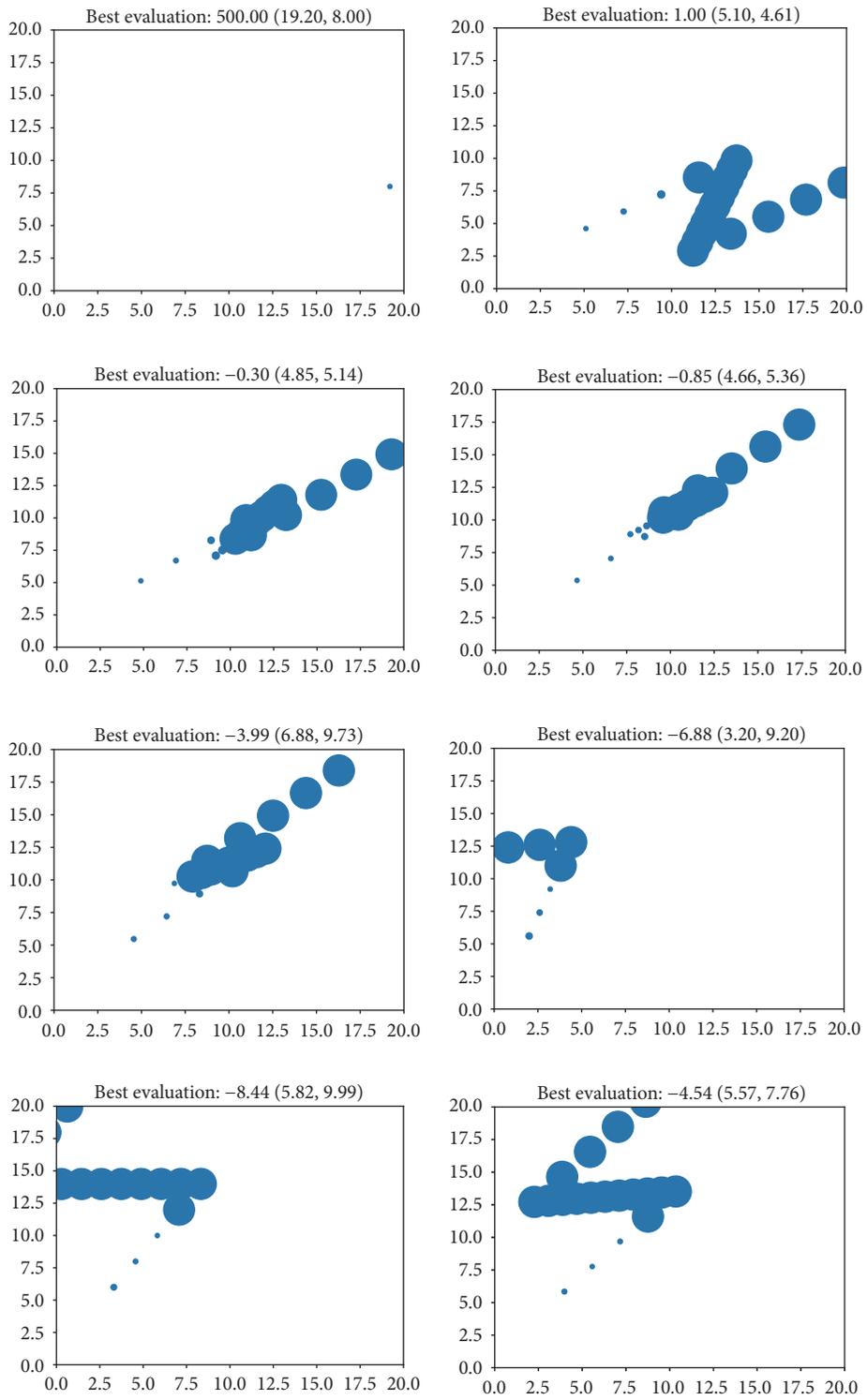


FIGURE 4: Cycles of SDN from left to right, top to bottom. Performance is indicated at the top of the plots, with the sizes of the evaluated locations inversely proportional to the value of the loss function (and scaled per plot). The figure shows that the smallest loss value, in a 2% vicinity of the minimum, was found as early as in the 7th cycle, without recourse to any kind of gradient value.

References

- [1] P. Baranyi, A. Csapo, and G. Sallai, "Cognitive infocommunications (CogInfoCom)," *Cognitive Infocommunications (CogInfoCom)*, pp. 1–219, 2015.
- [2] A. Csapo and P. Baranyi, "The spiral discovery method: An interpretable tuning model for CogInfoCom channels," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 16, no. 2, pp. 358–367, 2012.
- [3] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.
- [4] H. Takagi and H. Iba, "Preface interactive evolutionary computation," *New Generation Computing*, vol. 23, no. 2, pp. 113–114, 2005.
- [5] S. Shalev-Shwartz, O. Shamir, and S. Shammah, *Failures of deep learning*, 2017, arXiv preprint arXiv:1703.07950.
- [6] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87 of *Applied Optimization*, Springer, Amsterdam, The Netherlands, 2004.
- [7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [8] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.
- [9] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pp. 2176–2184, usa, June 2013.
- [10] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2121–2159, 2011.
- [11] D. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2014, arXiv preprint arXiv:1412.6980.
- [12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML '15)*, pp. 448–456, July 2015.
- [13] A. Neelakantan, L. Vilnis, Q. V. Le et al., *Adding gradient noise improves learning for very deep networks*, 2015, arXiv preprint arXiv:1511.06807.
- [14] P. Baranyi, "TP model transformation as a way to LMI-based controller design," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 2, pp. 387–400, 2004.
- [15] P. Baranyi, Y. Yam, and P. Várlaki, *Tensor product model transformation in polytopic model-based control*, CRC Press, 2013.
- [16] P. Baranyi, *TP-Model Transformation-Based-Control Design Frameworks*, Springer International Publishing, 2016.
- [17] L. de Lathauwer, B. de Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [18] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Huffel, "Dimensionality reduction for higher-order tensors: algorithms and applications," *International Journal of Pure and Applied Mathematics*, vol. 42, no. 3, pp. 337–343, 2008.
- [19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [20] N. Kalchbrenner, I. Danihelka, and A. Graves, *Grid long short-term memory*, 2015, arXiv preprint arXiv:1507.01526.
- [21] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Springer US, Boston, MA, USA, 2011.
- [22] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [23] L. Davis, *Handbook of genetic algorithms*, 1991.
- [24] M. Gen and R. Cheng, *Genetic algorithms and engineering optimization*, John Wiley & Sons, 2000.
- [25] J. H. Holland, *Complexity: A Very Short Introduction*, Oxford University Press, 2014.

Research Article

Effective Evolutionary Multilabel Feature Selection under a Budget Constraint

Jaesung Lee , Wangduk Seo , and Dae-Won Kim 

School of Computer Science and Engineering, Chung-Ang University, 221 Heukseok-dong, Dongjak-gu, Seoul 06974, Republic of Korea

Correspondence should be addressed to Dae-Won Kim; dwkim@cau.ac.kr

Received 29 September 2017; Accepted 21 January 2018; Published 7 March 2018

Academic Editor: Kevin Wong

Copyright © 2018 Jaesung Lee 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.

Multilabel feature selection involves the selection of relevant features from multilabeled datasets, resulting in improved multilabel learning accuracy. Evolutionary search-based multilabel feature selection methods have proved useful for identifying a compact feature subset by successfully improving the accuracy of multilabel classification. However, conventional methods frequently violate budget constraints or result in inefficient searches due to ineffective exploration of important features. In this paper, we present an effective evolutionary search-based feature selection method for multilabel classification with a budget constraint. The proposed method employs a novel exploration operation to enhance the search capabilities of a traditional genetic search, resulting in improved multilabel classification. Empirical studies using 20 real-world datasets demonstrate that the proposed method outperforms conventional multilabel feature selection methods.

1. Introduction

Multilabel classification has emerged as a promising technique for various applications, including lifelong structure monitoring [1], functional proteomics [2], and sentiment analysis [3]. These applications produce a series of labels for describing complicated concepts, which are compounded when high-level concepts are composed of multiple subconcepts, such as the environmental and operational conditions of structures [1, 4, 5]. Let $W \subset \mathbb{R}^d$ denote a set of patterns constructed from a set of features F . Then, each pattern $w_i \in W$, where $1 \leq i \leq |W|$, is assigned to a certain label subset $\lambda_i \subseteq L$, where $L = \{l_1, \dots, l_{|L|}\}$ and is a finite set of labels. Therefore, the task of multilabel classification is to identify a function that maps given instances into one of $2^{|L|}$ label subsets based on input feature values.

In practice, there can be a maximum number of features allowed because of the limits on data acquisition rates or energy consumption [6–8]. In reality, for example, this problem can emerge from the music applications on lightweight mobile devices. Applications for mobile devices typically have a limitation in computational capacity and there is a maximum number of allowed features to be extracted [9, 10].

This is because an overly excessive number of extracted features on mobile devices causes consumers to suffer low quality user experience due to unacceptable waiting or battery consumption.

Given input data with an original feature set F and label set L , the goal of our multilabel feature selection problem is to identify a feature subset $S \subset F$ with the maximum number of features n that yields the best multilabel classification accuracy [11, 12]. This problem is known as budgeted feature selection [13] or feature selection with test cost constraints [8, 14, 15]. However, most studies have been conducted from the perspective of traditional single-label learning. It should be noted, especially when a given constraint n is small, that our multilabel feature selection problem becomes more challenging in terms of classification accuracy due to the fact that a small number of features must support multiple labels simultaneously [16–19].

Multilabel feature selection methods can be categorized according to how they assess the importance of candidate feature subsets [16, 20–22]. Filter-based multilabel feature selection methods identify a final feature subset by focusing on the intrinsic discriminative power of features [21, 23–25]. Some multilabel learning algorithms have a feature

selection process embedded in their learning process [26, 27]. In contrast, wrapper-based multilabel feature selection methods assess the importance of feature subsets through a search process by using a multilabel classifier directly. This typically results in better classification accuracy [11, 12]. For this reason, we focus on a multilabel feature wrapper based on an evolutionary search process [28].

During the search process, each chromosome represents a feature subset and selects a number of features less than or equal to n . As a result, most features remain unselected by any chromosome in the population. This can lead to an ineffective search because important features can be continuously neglected. Without negatively affecting the strength of the evolutionary search, this problem can be solved by adding additional chromosomes that convey promising unselected features to the population. In this study, we propose an effective multilabel feature wrapper while considering the constraint of feature subset size. Experimental results demonstrate that the proposed method is able to identify an effective feature subset for multilabel classification with the aid of an enhanced evolutionary search process.

2. Related Work

In traditional single-label feature selection, the budgeted feature selection problem is treated as a special case of the feature selection problem where the algorithm should consider the effectiveness of the feature subset and the acquisition cost for gathering each feature simultaneously. To solve this problem, Zhang et al. [29] proposed a feature selection algorithm based on the bare bones particle swarm optimization, which considers the complexity of an algorithm due to additional parameters. Because the acquisition cost for each feature can be unequal, multiobjective particle swarm optimization approach for cost-based feature selection and return-cost-based binary firefly algorithm for feature selection are also studied [30, 31] which have another objective function of minimizing the cost sum of features.

In multilabel feature selection studies, one of the major trends is the application of a feature selection method for single-label problems by transforming multilabel datasets into single-label datasets [32, 33]. Although this strategy facilitates the use of conventional methods, which has advantages in terms of ease of use [34], algorithm adaptation strategies that directly manage multilabel problems have also been considered [35]. In these approaches, which are largely filter-based, a feature subset is obtained by optimizing a specific criterion, such as a joint learning criterion that involves simultaneous feature selection and multilabel learning [27, 36], $l_{2,1}$ -norm function optimization [37], label ranking error [26], Hilbert-Schmidt independence criterion [23], F -statistics [21], or mutual information [16, 24, 38]. However, these methods commonly suffer from low multilabel classification accuracy because of a lack of interaction with multilabel classifiers.

As a notable multilabel feature wrapper study, Zhang et al. [12] proposed a multilabel feature selection method based on a genetic algorithm (GA), which is the most common choice

in evolutionary feature wrapper studies [28]. Specifically, their method combined instance- and label-based evaluation metrics [39] as a fitness function to determine label dependency. However, in the original proposal, a maximum number of features to be selected were not considered during the genetic search process. The multilabel classification performance when considering the number of features to be selected was later demonstrated for comparison purposes [11]. During initialization, this method creates chromosomes by selecting a number of features less than n . During the genetic search process, this constraint is continuously satisfied by employing restrictive crossover and mutation operators [40] that immediately discard features randomly if the number of selected features exceeds n . Although this method satisfies the constraint, important features may be discarded, resulting in an ineffective feature subset.

Recent multilabel feature wrapper methods have treated the number of features to be selected as a secondary objective to be achieved by the evolutionary search process (i.e., multiobjective optimization [28]). This is achieved through a specifically designed ranking method for multiobjective optimization problems, known as nondominated sort [41], where the rank of each chromosome is based on the number of times it dominates other chromosomes in terms of two fitness values: multilabel classification accuracy and the number of selected features. Because the ranking of the chromosomes can be determined, it can be directly used in the natural selection process of a GA. Although the most common approach using a nondominated sorting method is NSGA-II [42], nondominated sorting has also been employed in other evolutionary search methods, including particle swarm optimization (PSO) [43]. A common drawback in these methods is that no solution may satisfy the feature number constraint if such a solution is not included in the final Pareto front. Additionally, they may suffer from unnecessary searches of infeasible solutions conveying unacceptable number of features.

Our review indicates that conventional multilabel feature wrappers can fail to identify a final solution that satisfies a given constraint. To remedy this limitation, in addition to the evolutionary process, it is necessary to devise a new process, namely, exploration operation, to find important features in a large set of novel features with the aid of an effective filter and supply them to the population to enhance the evolutionary search process. We summarize subsequent issues and corresponding reasons to our approach as follows.

- (i) The exploration operation must be able to identify promising features in a large unselected feature set size of $O(|F| - n) = O(|F|)$. To achieve this, we employ a criterion that measures the relevance score of features.
- (ii) The exploration operation must be computationally efficient to circumvent performance degradation of the entire search process. To achieve this, we employ a multilabel feature filter that is confirmed to be efficient because it only requires the dependency between two variables [16].
- (iii) Our exploration operation is designed to incur no additional parameter that may cause complicated

```

(1) procedure PROPOSED ALGORITHM( $v, m$ )                                ▷ allowed FFC  $v$ 
(2)    $t \leftarrow 0, u \leftarrow 0$                                     ▷  $t$ -th generation
(3)   initializing  $P(t)$                                            ▷ population  $P$  of  $t$ -th generation
(4)   evaluating  $P(t)$ 
(5)   While  $u \leq v$  do                                           ▷ if spent FFC  $u$  is less than  $v$ 
(6)     create  $G(t)$  using genetic operators
(7)     create  $E(t)$  using exploration operator based on  $G(t)$ 
(8)      $N(t) \leftarrow \{G(t) \cup E(t)\}$                             ▷ offspring set  $N(t)$ 
(9)     evaluate  $N(t)$  using a multi-label classifier
(10)    add  $N(t)$  to  $P(t)$ 
(11)     $t \leftarrow t + 1$ 
(12)    select  $P(t)$  from  $P(t - 1)$                                 ▷ natural selection
(13)     $u \leftarrow m + 2 \cdot |G(t)| \cdot t$                     ▷ update  $u$  based on spent FFC
(14)  end while
(15) end procedure

```

ALGORITHM 1: Procedures of proposed multilabel feature wrapper.

parameter control issues and increase the overall complexity of the algorithm [11, 44]. Based on the number of features given by the evolutionary search, it automatically identifies an effective feature subset that is composed only of novel features.

3. Proposed Method

3.1. Motivation and Approach. In this study, we enhance the performance of a population-based search, such as a GA, for multilabel feature selection with a budget constraint by introducing novel chromosomes that inject promising unselected features into the population. Figure 1 reveals several key issues that should be considered when introducing novel features into the evolutionary search-based multilabel feature selection process with a budget constraint. In the original feature set F , there may be a subset of important features that are strongly dependent on multiple labels, leading to excellent discriminative power in the multilabel classifier if they are included in the final feature subset. After a random initialization process is completed, important features, such as f_1 , may be unselected by any chromosome (feature subset) because each chromosome only covers a small number of features under the budget constraint n . It should be noted that $\lceil |F|/n \rceil$ chromosomes should be evaluated to consider all the features at least once, even though all chromosomes are forced to select disjoint feature subsets, which incurs an expensive computational cost. Instead, the proposed method identifies promising features with the help of the employed filter without explicit evaluation of candidate feature subsets.

Next, genetic operators, such as crossovers and mutations, are applied to the population to create new chromosomes. However, unselected important features may not be considered because new chromosomes are created by exchanging the alleles of their ancestors. This means that if ancestors commonly unselect a feature, then their offspring will also unselect that feature. The only chance to add neglected features into the offspring creation process is

through the use of a mutation operation. However, this is computationally inefficient because the mutation operation is done by selecting features randomly and, additionally, the mutation rate is set to a small value in order to achieve the convergence. Thus, a large number of iterations or generations should be spent to introduce important features into the population randomly.

In the proposed method, the exploration operator is applied to each of the new offspring to create novel chromosomes that contain promising features that were not considered by the original offspring. During each exploration operation, we calculate the dependency of unselected features on multiple labels (l_1, l_2, \dots, l_8). After the ranking of each feature is computed (e.g., $f_1 \rightarrow f_{44} \rightarrow f_{32} \rightarrow f_3 \rightarrow \dots$), a new chromosome that selects the most promising features is created. Finally, exploration and genetic operation-based chromosomes are then merged into a single population.

This paper presents an effective evolutionary search method that remedies the aforementioned issues. In Section 3.2, we discuss the procedural steps of the proposed method and how to handle the issues associated with the exploration operation and the creation of new chromosomes. Section 3.3 presents a mutual-information-based search method for efficiently capturing the relationships between features and labels.

3.2. Algorithm. Algorithm 1 outlines the pseudocode for the procedures used in the proposed method. The terms used for describing the algorithm are summarized in “Terms Used in This Study and Meanings” section. The feature selection vector in a chromosome is a binary string where each bit represents an individual feature, with values of one and zero representing selected and unselected features, respectively. In the initialization step (line (3)), the algorithm generates m chromosomes via random assignment of maximum n binary bits. The selected feature subset (S_c) encoded in $c \in P(t)$ is then evaluated using a fitness function. We use multilabel classification error as the fitness function for the selected

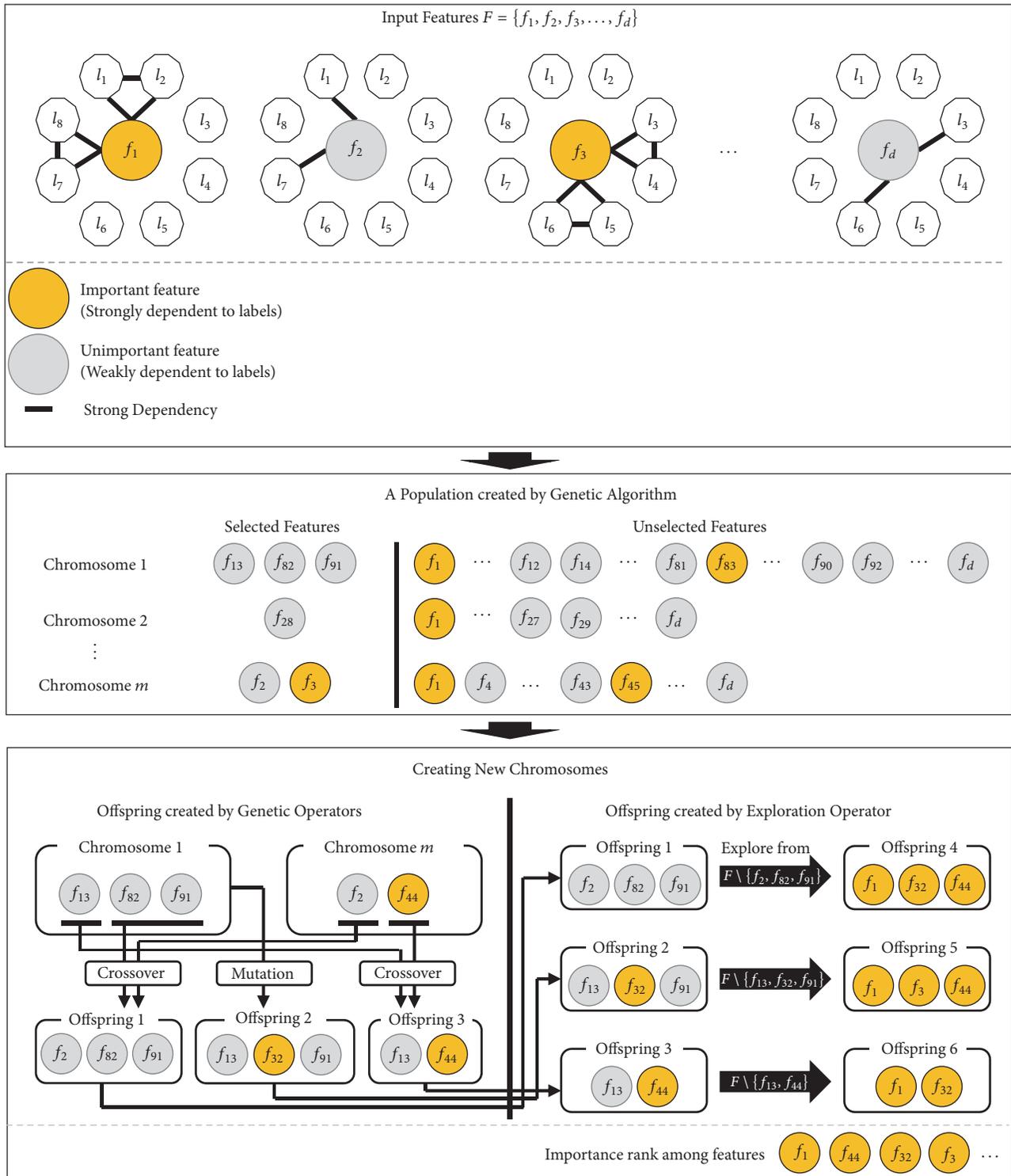


FIGURE 1: The cooperation process between genetic and exploration operation.

feature subset. Because m chromosomes must be evaluated in order to obtain their fitness values, m fitness function calls (FFCs) are used in line (4).

After performing the initialization process, the proposed method performs a reproduction process that can be

divided into two parts: reproduction via genetic operators and reproduction via the exploration operator. First, the proposed method creates an offspring set $G(t)$ (line (6)) using restrictive crossover and mutation operators to control the number of selected features [40]. Next, the exploration

```

(1) procedure EXPLORE( $G(t)$ )
(2)    $E(t) \leftarrow \{\emptyset\}$ 
(3)   for each  $c \in G(t)$  do
(4)      $Z \leftarrow \{\emptyset\}$  ▷ initialize novel feature subset  $Z$ 
(5)     for  $i = 1$  to  $|S_c|$  do ▷ feature subset selected by  $c, S_c$ 
(6)       find the best feature  $f^+ = \arg \max_{f^+ \in \{F \setminus \{S_c \cup Z\}\}} Q(f^+, L)$ 
(7)       add  $f^+$  to  $Z$ 
(8)     end for
(9)     add  $Z$  to  $E(t)$  as a chromosome
(10)  end for
(11) end procedure

```

ALGORITHM 2: Procedures of exploration operator.

operator identifies unselected promising features from the perspective of each chromosome in $G(t)$ and encodes them into a new chromosome in $E(t)$ (line (7)). For balance between the genetic and exploration operations, we set the size of $E(t)$ to the same value as that of $G(t)$ because $E(t)$ must be evaluated in order to determine its fitness. These two sets of chromosomes are then combined to form the offspring set $N(t)$ of the t th population (line (8)). To evaluate the fitness of the offspring set, the proposed method uses a certain number of FFCs (line (9)). Specifically, the proposed method uses $2 \cdot |G(t)|$ FFCs in one generation. Next, $N(t)$ is added to $P(t)$ and m chromosomes with higher fitness values are selected (line (11)). This procedure is repeated until the algorithm uses all of its allowed FFCs. This limit is denoted ν and is chosen by the user. The output of Algorithm 1 is the best feature subset obtained during evolution.

3.3. Exploration Operator. Because a feature subset selects a small number of features within n and most features will remain unselected, the exploration operator is needed in order to explore a large set of unselected features. Algorithm 2 outlines the pseudocode for the proposed exploration operator. For each offspring generated by the genetic operators, we iteratively select relevant features that maximize the objective function and that were not selected by the offspring c until the subset size becomes $|S_c|$, where $|S_c|$ is the subset size of c . Thus, proposed exploration operation does not incur additional parameter for determining the number of features to be selected.

To implement our exploration operation, we employ an effective filter method called the scalable criterion for large label sets (SCLS) [16] as an objective function $Q(f^+, L)$, where L is the label set. The selection of the i th feature from the set $\{F \setminus \{S_c \cup Z\}\}$, where Z is a feature subset with $i - 1$ features when selecting i th feature, is performed by identifying f_i that maximizes the value of the following relevance evaluation [17]:

$$\max_{f_i \in \{F \setminus \{S_c \cup Z\}\}} [D(f_i) - R(f_i)], \quad (1)$$

where $D(f_i)$ and $R(f_i)$ denote the dependency of f_i on L and the dependency of f_i on the selected features of

Z , respectively. From [17], (1) can be reformulated as follows:

$$\max_{f_i \in \{F \setminus \{S_c \cup Z\}\}} \left[\sum_{l \in L} M(f_i; l) - \sum_{f \in S_{i-1}} M(f_i; f) \right], \quad (2)$$

where $M(x; y) = H(x) - H(x, y) + H(y)$ is the mutual information between variables x and y and $H(x) = -\sum P(x) \log P(x)$ is the joint entropy of the probability functions $P(x)$, $P(y)$, and $P(x, y)$. Following from (2), $D(f_2)$ can be calculated as follows:

$$D(f_2) = \sum_{l \in L} M(f_2; l). \quad (3)$$

As (2), $R(f_2)$ can be calculated as

$$R(f_2) = \sum_{l \in L} M(f_2; l). \quad (4)$$

In order to calculate $R(f_2)$ while considering adaptability against the scaling of $D(f_2)$ and avoiding repetitive calculations by $f \in S$ and $l \in L$, let $\text{Red}(f_2)$ be represented as follows:

$$R(f_2) = \alpha \cdot D(f_2) = \alpha \sum_{l \in L} M(f_2; l), \quad (5)$$

where $0 \leq \alpha \leq 1$, which must be estimated, determines the reduction with relevance to f_2 based on $D(f_2)$, while circumventing the repetitive calculations for reduction against each label. According to [16], α can be approximated as follows:

$$\alpha \approx \frac{M(f_2; f_1)}{H(f_2)}. \quad (6)$$

As a result, the relevance evaluation for f_2 is performed as follows:

$$J = \sum_{l \in L} M(f_2; l) - \frac{M(f_2; f_1)}{H(f_2)} \sum_{l \in L} M(f_2; l). \quad (7)$$

Equation (7) represents how the relevance evaluation can be performed when $i = 2$. By considering the previously

TABLE 1: Standard characteristics of employed datasets.

Dataset	$ W $	$ F $	Type	$ L $	Card.	Den.	Distinct.	Domain
Birds	645	260	Mixed	19	1.014	0.053	133	Audio
Emotions	593	72	Numeric	6	1.869	0.311	27	Music
Enron	1,702	1,001	Nominal	53	3.378	0.064	753	Text
Genbase	662	1,185	Nominal	27	1.252	0.046	32	Biology
LLog	1,460	1,004	Nominal	75	1.180	0.016	304	Text
Mediamill	43,907	120	Numeric	45	1.245	0.028	94	Video
Medical	978	1,494	Nominal	45	1.245	0.028	94	Text
Scene	2,407	294	Numeric	6	1.074	0.179	15	Images
Slashdot	3,782	1,079	Nominal	22	1.181	0.054	156	Text
TMC2007	28,596	981	Numeric	22	2.158	0.098	1,341	Text
Yeast	2,417	103	Numeric	14	4.237	0.303	198	Biology
Arts	7,484	1,157	Numeric	26	1.654	0.064	599	Text
Business	11,214	1,096	Numeric	30	1.599	0.053	233	Text
Computers	12,444	1,705	Numeric	33	1.507	0.046	428	Text
Education	12,030	1,377	Numeric	33	1.463	0.044	511	Text
Entertain	12,730	1,600	Numeric	21	1.414	0.067	337	Text
Health	9,205	1,530	Numeric	32	1.644	0.051	335	Text
Reference	8,027	1,984	Numeric	33	1.174	0.036	275	Text
Science	6,428	1,859	Numeric	40	1.450	0.036	457	Text
Social	12,111	2,618	Numeric	39	1.279	0.033	361	Text
Society	14,512	1,590	Numeric	27	1.670	0.062	1,054	Text

selected features in Z , the final relevance evaluation can be represented as follows:

$$\max_{f_i \in \{F \setminus \{S_c \cup Z\}\}} \left[\sum_{l \in L} M(f_i; l) - \sum_{f \in S_{i-1}} \sum_{l \in L} \frac{M(f_i; l)}{H(f_i)} M(f_i; f) \right]. \quad (8)$$

Equation (8) is the objective function for selecting relevant features from the unselected feature subset used by our exploration operation.

3.4. Experimental Settings. We experimented on 20 different datasets from various domains. The Birds dataset is audio data containing examples of multiple bird calls. The Emotions dataset is music data classified into six emotional clusters. The Enron, Language Log (LLog), and Slashdot datasets were generated from text mining applications, where each feature corresponds to the occurrence of a word and each label represents the relevancy of each text pattern to a specific subject. The Genbase and Yeast datasets come from the biological domain and include information about the functions of genes and proteins. The Mediamill dataset is video data from an automatic detection system. The Medical dataset was sampled from a large corpus of suicide letters obtained from the natural language processing of clinical free text. The Scene dataset is related to the semantic indexing of still scenes, where each scene may contain multiple objects. The TMC2007 dataset contains safety reports of complex space system. The remaining nine datasets come from the Yahoo dataset collection. We performed unsupervised dimensionality reduction on text datasets, including the TMC2007 and Yahoo collections, which were composed of more than

10,000 features. Specifically, the top 2% and 5% of features with the highest document frequency were retained for TMC2007 and the Yahoo datasets, respectively [45]. In the text mining domain, existing studies report that classification performance will not suffer significantly from the retention of 1% of features based on document frequency [46].

Table 1 contains the standard statistics for the multilabel datasets employed in our experiments, including the number of patterns in the dataset $|W|$, number of features $|F|$, type of features, and number of labels $|L|$. When the feature type is numeric, we discretize the features by using the supervised discretization method [47] for multilabel naive Bayes classifier (MLNB) [12]. Specifically, each observed numeric value is assigned to one of several bins that are automatically determined by using the discretization method. The label cardinality *Card* represents the average number of labels for each instance. The label density *Den* is the label cardinality over the total number of labels. The number of distinct label sets *Distinct* indicates the number of unique label subsets in L . *Domain* represents the application that each dataset was extracted from.

We measured the mean size of the selected feature subsets for both the proposed method and the conventional multilabel feature selection methods (GA with restrictive genetic operators [40] (RGA), NSGA-II [43], and MPSOFS [43]) to determine which methods achieved to select less than 10 features. Specifically, we provide detailed parameter setting to support good reproducibility as follows:

- (i) RGA creates $m = 20$ initial solutions by selecting less than $n = 10$ features randomly in accordance with each chromosome. Each solution in the initial

population $P(t)$, where $t = 0$, is evaluated using an employed multilabel classifier. Next, the RGA creates an offspring set $N(t)$ by using genetic operators. To apply the crossover operator, two solutions in $P(t)$ are randomly selected and mated; thereafter, one solution in $P(t)$ is randomly selected and mutated. In this study, we employed restrictive crossover and restrictive mutation operators with both crossover rate and mutation rate set to 1.0. Therefore, for each iteration, the GA creates three new solutions to compose $N(t)$. Each newly created solution is evaluated using the multilabel classifier. To create $P(t + 1)$, $N(t)$ is added to $P(t)$, and 20 solutions with higher fitness values are selected. This procedure is repeated until the RGA spends 100 FFCs.

- (ii) NSGA-II creates $m = 20$ initial solutions randomly, the same number RGA creates. The maximum number of allowed feature is set to $|F|$ because the NSGA-II naturally minimizes the number of selected features. Each solution in $P(t)$ is evaluated using an employed multilabel classifier and the number of features. The NSGA-II then creates $N(t)$ where $|N(t)| = 3$ which is the same setting of RGA. To create $P(t + 1)$, $N(t)$ is added to $P(t)$, and the superiority of each solution is determined by the nondominated sort method. After the superiority among solutions in $\{P(t) \cup N(t)\}$ is determined, the top 20 solutions are selected to form $P(t + 1)$. This procedure is repeated until the NSGA-II spends 100 FFCs.
- (iii) MPSOFS creates 20 initial solutions randomly, the same number RGA creates. Each solution in $P(t)$ is evaluated using an employed multilabel classifier and the number of features and ranked using the nondominated sort method. The MPSOFS then preserves the best solution of $P(t)$ called the global best solution. In addition, the best solution which each chromosome experienced is also preserved; this is called the individual best solution, and therefore there are 20 individual best solutions. Thereafter, the MPSOFS updates the representation of each chromosome based on the global best solution and its own individual best solution using a velocity with inertia weight of 0.7298 and two acceleration coefficients of 1.4962 suggested from the study of [48]. After all chromosomes in $P(t)$ are modified, they are evaluated and regarded as $P(t + 1)$. This procedure is repeated until the MPSOFS spends 100 FFCs.

Although different parameter setting may result in better performance, we fixed the size of the population m to 20 and the number of spent FFCs ν to 100 for all the methods to ensure a fair comparison. To evaluate the quality of the feature subsets obtained by each method, we used MLNB classifier because it outputs a predicted label subset based on the intrinsic characteristics of a given dataset without requiring any complicated parameter-tuning process that might influence the final multilabel classification performance [39]. For the sake of fairness, we used the hold-out cross-validation method for each experiment [11, 49]. 80% of the samples in a

given dataset were randomly chosen as the training set for multilabel feature selection and classifier training, while the remaining 20% of the samples were used as the test set to obtain the multilabel classification performance. For both the RGA and the proposed method, we set the population size to 20 and the maximum number of allowed FFCs to 100. Each experiment was repeated 10 times and the average value was used to represent the classification performance of each feature selection method.

We employed four evaluation metrics: Hamming loss, multilabel accuracy, ranking loss, and normalized coverage. Let $T = \{(T_i, \lambda_i) \mid 1 \leq i \leq |T|\}$ be a given test set where $\lambda_i \subseteq L$ is a correct label subset. For a given test sample T_i , a classifier, such as MLNB, should output a set of confidence values $0 \leq \psi_{i,l} \leq 1$ for each label $l \in L$. If a confidence value $\psi_{i,l}$ is larger than a predefined threshold value, such as 0.5, the corresponding label l will be included in the predicted label subset Y_i . Based on the ground truth λ_i , confidence values $\psi_{i,l}$, and predicted label subset Y_i , multilabel classification performance can be measured with each evaluation metric [33, 45, 50].

Multilabel accuracy is defined as follows:

$$\text{mlacc}(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{|\lambda_i \cap Y_i|}{|\lambda_i \cup Y_i|}. \quad (9)$$

Hamming loss is defined as follows:

$$\text{hloss}(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{1}{|L|} |\lambda_i \Delta Y_i|, \quad (10)$$

where Δ denotes the symmetric difference between two sets. Ranking loss is defined as follows:

$$\text{rloss}(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{|\{(a, b) \mid a \in \lambda_i, b \in \bar{\lambda}_i, \psi_{i,a} \leq \psi_{i,b}\}|}{|\lambda_i| |\bar{\lambda}_i|}, \quad (11)$$

where $\bar{\lambda}_i$ is a complementary set of λ_i . Therefore, ranking loss measures the average fraction of (a, b) pairs with $\psi_{i,a} \leq \psi_{i,b}$ over all possible relevant and irrelevant label pairs. Finally, normalized coverage is defined as follows:

$$\text{ncov}(T) = \frac{1}{|L|} \left(\frac{1}{|T|} \sum_{i=1}^{|T|} \max_{l \in \lambda_i} \text{rank}(l) - 1 \right), \quad (12)$$

where $\text{rank}(\cdot)$ returns the rank of the corresponding relevant label $l \in \lambda_i$ according to $\psi_{i,l}$ in nonincreasing order. Therefore, normalized coverage measures how many labels must be marked as positive for all relevant labels to be positive. Higher values of multilabel accuracy and lower values of Hamming loss, ranking loss, and normalized coverage indicate good classification performance.

Additionally, because we are interested in the superiority of the proposed method over conventional multilabel feature selection methods, we perform the Wilcoxon signed-rank test [51] to validate the performance of the proposed method. Let d_i be the difference between the performance of the two methods for the i th dataset. The differences are ranked

TABLE 2: Comparison results for multilabel feature selection methods in terms of selected feature subset size (mean \pm std. deviation). The \times symbol is used to indicate that the corresponding method failed to select less than 10 features for the dataset.

Dataset	Proposed	RGA	NSGA-II	MPSOFS
Birds	7 \pm 2	5 \pm 2	99 \pm 51 \times	138 \pm 4 \times
Emotions	8 \pm 1	9 \pm 1	50 \pm 6 \times	37 \pm 4 \times
Enron	8 \pm 1	9 \pm 1	74 \pm 51 \times	527 \pm 23 \times
Genbase	9 \pm 0	7 \pm 1	974 \pm 139 \times	637 \pm 24 \times
LLog	8 \pm 1	7 \pm 2	205 \pm 108 \times	522 \pm 17 \times
Mediamill	5 \pm 1	4 \pm 0	7 \pm 2	52 \pm 3 \times
Medical	8 \pm 1	8 \pm 1	664 \pm 138 \times	762 \pm 30 \times
Scene	9 \pm 1	9 \pm 0	137 \pm 31 \times	147 \pm 4 \times
Slashdot	9 \pm 0	8 \pm 2	970 \pm 85 \times	569 \pm 22 \times
TMC2007	8 \pm 1	8 \pm 1	495 \pm 93 \times	506 \pm 20 \times
Yeast	9 \pm 1	9 \pm 1	34 \pm 10 \times	52 \pm 3 \times
Arts	9 \pm 1	7 \pm 1	1,019 \pm 85 \times	613 \pm 25 \times
Business	4 \pm 2	4 \pm 2	130 \pm 140 \times	578 \pm 20 \times
Education	8 \pm 1	8 \pm 2	1263 \pm 51 \times	742 \pm 24 \times
Entertainment	9 \pm 0	8 \pm 2	985 \pm 227 \times	840 \pm 28 \times
Health	7 \pm 1	4 \pm 1	842 \pm 184 \times	814 \pm 33 \times
Reference	7 \pm 2	8 \pm 1	1052 \pm 334 \times	1058 \pm 42 \times
Science	9 \pm 0	7 \pm 1	969 \pm 265 \times	993 \pm 65 \times
Social	6 \pm 1	8 \pm 1	1,699 \pm 401 \times	1,353 \pm 71 \times
Society	7 \pm 2	4 \pm 3	498 \pm 95 \times	826 \pm 28 \times

based on their absolute values and the smallest d_i is assigned to the first rank. If ties occur, average ranks are assigned. Let R^+ be the sum of the ranks for the datasets on which the compared method outperforms the proposed method, defined as follows:

$$R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i). \quad (13)$$

Let R^- be the sum of the ranks for the datasets on which the proposed method outperforms the compared method. Then, based on the critical values from the Wilcoxon test, for a confidence level of $\alpha = 0.05$ and $N = 20$, the difference between the compared methods is significant if $\min(R^+, R^-)$ is less than or equal to 8. In this case, the null hypothesis of equal performance is rejected.

4. Experimental Results

4.1. Comparison Results. Table 2 contains the results for the mean size and standard deviation of the selected feature subsets of the proposed method and conventional multilabel feature selection methods when the evaluation metric is multilabel accuracy. The \times symbol indicates methods that failed to satisfy given constraint for the corresponding dataset. The proposed method and RGA both selected less than 10 features for all datasets. The NSGA-II and MPSOFS methods failed to select less than 10 features for all datasets other than the Mediamill dataset for NSGA-II, despite having objective functions to minimize feature subset sizes. Because the NSGA-II and MPSOFS failed to select less than 10 features for most datasets, we compared the performance of the

proposed method with the performance of the RGA from subsequent experiments. It should be noted that n can be set to a larger value than 10, such as 30 or 50. The experimental results in Table 2 show that the NSGA-II or MPSOFS will fail to satisfy the given constraints because they output the final feature subset, which is composed of tens or hundreds of features for most experiments.

Tables 3 and 4 contain the experimental results for the proposed method and RGA on 20 multilabel datasets, presented as the average performances for hold-out cross-validation with corresponding standard deviations. Table 3 contains the performance results for multilabel accuracy and Hamming loss, and Table 4 contains the performance results for ranking loss and normalized coverage. The best performance between the two methods is indicated by bold font and a \checkmark symbol. Finally, Table 5 contains the results of the Wilcoxon signed-rank test for the proposed method against RGA for Genbase dataset with a significance threshold of $\alpha = 0.05$. For each evaluation metric, the winner of each comparison is indicated with bold font and the corresponding sum of the outperformed rank R^+ over the total rank and p values are presented in the parenthesis. We observed a similar tendency from the same experiments on the other multilabel datasets.

As shown in Tables 3 and 4, the proposed method outperformed RGA for most multilabel datasets. Specifically, the proposed method achieved the best performance for 90% of the datasets in terms of multilabel accuracy, 95% of the datasets in terms of Hamming loss, 95% of the datasets in terms of ranking loss, and 100% of the datasets in terms of normalized coverage. Thus, the proposed method

TABLE 3: Comparison results for multilabel feature selection methods in terms of multilabel accuracy and Hamming loss (mean \pm std. deviation). The \checkmark symbol indicates the method that achieves the best performance for each dataset.

Methods	Evaluation measure			
	Multi-label accuracy		Hamming loss	
	Proposed	RGA	Proposed	RGA
Birds	0.497 \pm 0.048 \checkmark	0.459 \pm 0.048	0.055 \pm 0.005 \checkmark	0.056 \pm 0.004
Emotions	0.460 \pm 0.020 \checkmark	0.447 \pm 0.029	0.243 \pm 0.022 \checkmark	0.252 \pm 0.016
Enron	0.360 \pm 0.021 \checkmark	0.271 \pm 0.042	0.056 \pm 0.002 \checkmark	0.060 \pm 0.001
Genbase	0.886 \pm 0.041 \checkmark	0.155 \pm 0.097	0.011 \pm 0.004 \checkmark	0.042 \pm 0.003
LLog	0.213 \pm 0.027 \checkmark	0.166 \pm 0.026	0.016 \pm 0.001	0.016 \pm 0.001 \checkmark
Mediamill	0.366 \pm 0.002 \checkmark	0.359 \pm 0.005	0.034 \pm 0.000	0.034 \pm 0.000 \checkmark
Medical	0.517 \pm 0.048 \checkmark	0.097 \pm 0.046	0.018 \pm 0.002 \checkmark	0.026 \pm 0.002
Scene	0.408 \pm 0.019 \checkmark	0.352 \pm 0.030	0.157 \pm 0.002	0.154 \pm 0.005 \checkmark
Slashdot	0.144 \pm 0.017 \checkmark	0.031 \pm 0.011	0.048 \pm 0.001 \checkmark	0.053 \pm 0.000
TMC2007	0.372 \pm 0.005 \checkmark	0.318 \pm 0.020	0.084 \pm 0.001 \checkmark	0.088 \pm 0.001
Yeast	0.465 \pm 0.013 \checkmark	0.442 \pm 0.019	0.224 \pm 0.006 \checkmark	0.225 \pm 0.010
Arts	0.140 \pm 0.012 \checkmark	0.049 \pm 0.015	0.060 \pm 0.001 \checkmark	0.063 \pm 0.001
Business	0.678 \pm 0.011	0.678 \pm 0.008 \checkmark	0.029 \pm 0.001	0.029 \pm 0.001 \checkmark
Education	0.109 \pm 0.019 \checkmark	0.033 \pm 0.011	0.042 \pm 0.001 \checkmark	0.044 \pm 0.001
Entertain	0.233 \pm 0.016 \checkmark	0.128 \pm 0.042	0.058 \pm 0.000 \checkmark	0.065 \pm 0.002
Health	0.510 \pm 0.018 \checkmark	0.402 \pm 0.016	0.040 \pm 0.001 \checkmark	0.049 \pm 0.001
Reference	0.382 \pm 0.044	0.393 \pm 0.011 \checkmark	0.030 \pm 0.001 \checkmark	0.034 \pm 0.001
Science	0.120 \pm 0.011 \checkmark	0.042 \pm 0.015	0.034 \pm 0.001 \checkmark	0.036 \pm 0.001
Social	0.546 \pm 0.018 \checkmark	0.134 \pm 0.060	0.024 \pm 0.001 \checkmark	0.030 \pm 0.001
Society	0.304 \pm 0.135 \checkmark	0.280 \pm 0.146	0.055 \pm 0.001 \checkmark	0.059 \pm 0.001

TABLE 4: Comparison results for multilabel feature selection methods in terms of ranking loss and normalized coverage (mean \pm std. deviation). The \checkmark symbol indicates the method that achieves the best performance for each dataset.

Methods	Evaluation measure			
	Ranking loss		Normalized coverage	
	Proposed	RGA	Proposed	RGA
Birds	0.143 \pm 0.015 \checkmark	0.166 \pm 0.019	0.227 \pm 0.019 \checkmark	0.248 \pm 0.028
Emotions	0.218 \pm 0.025	0.217 \pm 0.029 \checkmark	0.499 \pm 0.028 \checkmark	0.524 \pm 0.030
Enron	0.098 \pm 0.008 \checkmark	0.115 \pm 0.008	0.277 \pm 0.001 \checkmark	0.296 \pm 0.010
Genbase	0.035 \pm 0.026 \checkmark	0.152 \pm 0.037	0.084 \pm 0.026 \checkmark	0.212 \pm 0.029
LLog	0.170 \pm 0.019 \checkmark	0.179 \pm 0.021	0.215 \pm 0.024 \checkmark	0.223 \pm 0.022
Mediamill	0.057 \pm 0.001 \checkmark	0.058 \pm 0.001	0.194 \pm 0.003 \checkmark	0.197 \pm 0.002
Medical	0.093 \pm 0.026 \checkmark	0.173 \pm 0.260	0.132 \pm 0.027 \checkmark	0.199 \pm 0.023
Scene	0.159 \pm 0.012 \checkmark	0.188 \pm 0.015	0.311 \pm 0.007 \checkmark	0.326 \pm 0.012
Slashdot	0.247 \pm 0.004 \checkmark	0.297 \pm 0.010	0.301 \pm 0.004 \checkmark	0.353 \pm 0.010
TMC2007	0.113 \pm 0.004 \checkmark	0.154 \pm 0.006	0.254 \pm 0.004 \checkmark	0.316 \pm 0.008
Yeast	0.199 \pm 0.007 \checkmark	0.200 \pm 0.008	0.550 \pm 0.010 \checkmark	0.553 \pm 0.013
Arts	0.161 \pm 0.017 \checkmark	0.180 \pm 0.019	0.260 \pm 0.016 \checkmark	0.275 \pm 0.019
Business	0.059 \pm 0.025 \checkmark	0.062 \pm 0.025	0.129 \pm 0.024 \checkmark	0.132 \pm 0.023
Education	0.095 \pm 0.004 \checkmark	0.109 \pm 0.003	0.152 \pm 0.004 \checkmark	0.168 \pm 0.004
Entertain	0.130 \pm 0.005 \checkmark	0.137 \pm 0.005	0.215 \pm 0.005 \checkmark	0.222 \pm 0.009
Health	0.089 \pm 0.028 \checkmark	0.107 \pm 0.027	0.161 \pm 0.025 \checkmark	0.179 \pm 0.026
Reference	0.110 \pm 0.022 \checkmark	0.119 \pm 0.023	0.155 \pm 0.023 \checkmark	0.164 \pm 0.022
Science	0.138 \pm 0.005 \checkmark	0.152 \pm 0.003	0.201 \pm 0.007 \checkmark	0.213 \pm 0.005
Social	0.073 \pm 0.010 \checkmark	0.107 \pm 0.027	0.124 \pm 0.010 \checkmark	0.132 \pm 0.011
Society	0.143 \pm 0.005	0.137 \pm 0.005 \checkmark	0.249 \pm 0.006 \checkmark	0.261 \pm 0.005

TABLE 5: Wilcoxon signed-rank test results for the proposed method against RGA for Genbase dataset with a significance threshold of $\alpha = 0.05$, sum of outperformed rank R^+ over the total rank and p values.

Evaluation measures	Result	Proposed versus RGA	
		Stats	p value
Hamming loss	Win	55/55	$2.0e - 3$
Multilabel accuracy	Win	55/55	$2.0e - 3$
Ranking loss	Win	55/55	$2.0e - 3$
Normalized coverage	Win	55/55	$2.0e - 3$

significantly outperforms RGA for all evaluation metrics. This is evident from the experimental results shown in Table 5, which clearly demonstrate that the proposed method is statistically superior to RGA.

4.2. Analysis. Figure 2 shows the convergence behaviors of the GA and proposed method according to the number of spent FFCs (u) in terms of the multilabel accuracy; the horizontal axis represents u , and the vertical axis indicates the multilabel accuracy performance. Because the convergence behaviors may differ according to each experiment owing to the stochastic nature of the population-based search methods, we set the same initialized population in both algorithms and averaged the multilabel accuracy performance of the top elitist in the population after conducting the experiment 10 times. Figure 2 shows that the multilabel accuracy performance monotonically improves with u . Because the initialization steps consume 20 FFCs and the two methods have the same initialized population that is randomly created, both methods gradually improve the multilabel accuracy initially. However, the experimental results indicate that the multilabel accuracy value of the proposed method is dramatically improved when $u \geq 20$ because the exploration operator is applied to the population after the initialization. Thus, Figure 2 indicates that the proposed method can efficiently locate a good feature subset from unselected features.

The goal of our exploration operation introduces novel promising features that would effectively improve the multilabel classification performance. To validate the effectiveness of our exploration operation, we conduct an additional experiment by comparing the fitness values of the offspring set created by the proposed exploration operation and the random operation, respectively. Specifically, 50 chromosomes, namely, G , that select 10 or lesser number of features as the same initialization procedure of RGA were used and 50 new chromosomes are then created by applying the proposed exploration operation to each chromosome in G to form the first offspring set. Thereafter, for the sake of comparison, novel features with regard to each chromosome in G are selected randomly and introduced to create the second offspring set. Finally, the fitness values of the first and second offspring sets in terms of the four performance measures are measured. Figure 3 shows the box plots of fitness values given by the two offspring sets of the Genbase dataset. The experimental results indicate that the fitness values of the first offspring set (Proposed) is much better than that of the second offspring set (Random) from the viewpoint of all

measures, indicating that the proposed exploration operation has a much better search capability than the random search.

5. Conclusion

We proposed an effective evolutionary search-based feature selection method with a budget constraint for multilabel classification. As a feature subset selects a small number of features within the maximum allowed number of features and most features are unselected in the budget constraint problem, we employ a novel exploration operation to find relevant features in the large unselected feature subset. Our experiments on 20 real-world datasets demonstrated that proposed exploration operator successfully enhances the search capability of genetic search, resulting in an improvement in multilabel classification. The results also showed that the proposed method can search a feature subset successfully, which does not violate the budget constraint. Statistical tests showed that our method outperformed conventional methods in four performance measures. Although the proposed exploration operation improves the effectiveness of evolutionary search without incurring additional parameters, it cannot be applied directly to certain types of evolutionary search algorithms, such as particle swarm optimization, which do not depend on offspring sets. Thus, an additional consideration should be made to design a new exploration operation for such cases.

A future research direction will be a study on an evolutionary algorithm. The proposed method is a genetic algorithm based feature selection; however, it can be applied to other evolutionary algorithms such as the Estimation of Distribution Algorithm. We would like to study this issue further.

Terms Used in This Study and Meanings

Constants

- t : Number of generations
- m : The size of the population, $|P(t)| = m$
- n : Maximum number of allowed features selected by S_c
- c : A chromosome in $P(t)$
- S_c : A selected feature subset represented by c
- v : Maximum number of allowed fitness function calls (FFCs)
- u : Number of spent FFCs, $u = m + 2 \cdot |G(t)| \cdot t$.

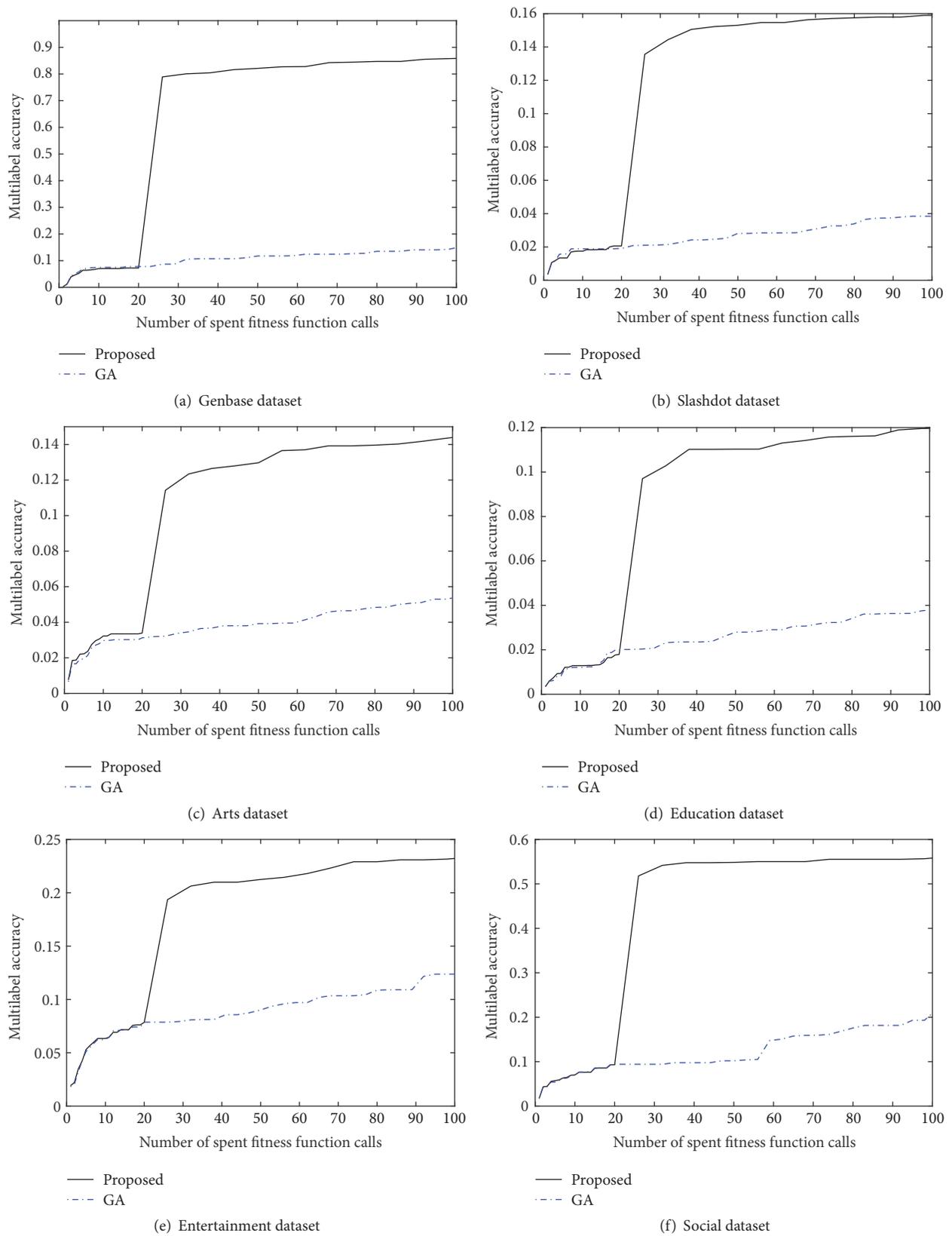


FIGURE 2: Comparison results of the convergence between RGA and the proposed method in terms of multilabel accuracy (a higher value indicates a good classification performance).

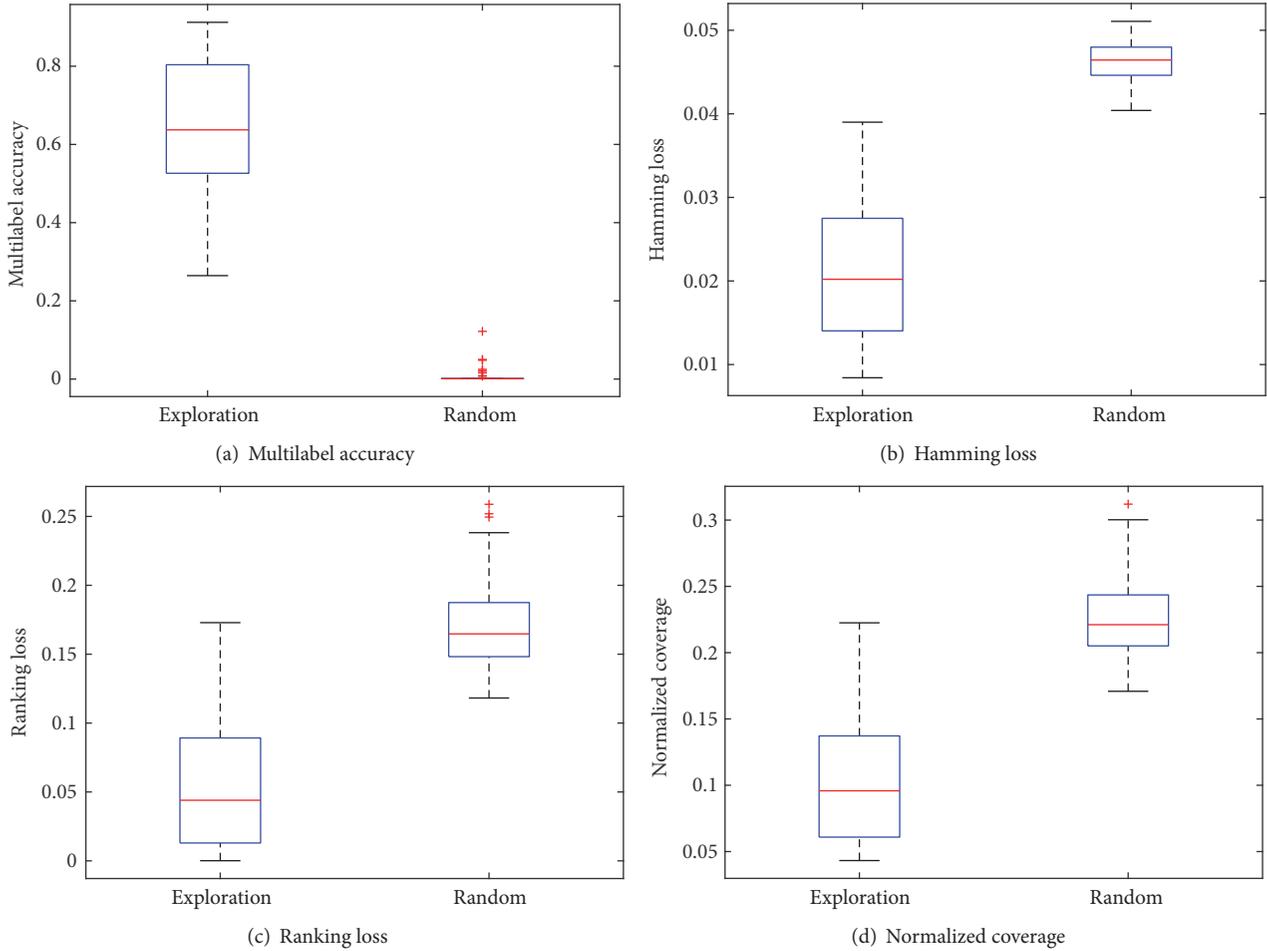


FIGURE 3: Comparison results showing the effectiveness of the proposed exploration operator and random search in terms of the four performance measures on the Genbase dataset.

Sets

$P(t)$: The population at the t th generation

$G(t)$: A set of newly created solutions from genetic operator

$E(t)$: A set of newly created solutions from exploration operator

$N(t)$: A set of newly created solutions from $P(t)$, $G(t) \cup E(t)$.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Chung-Ang University Research Grants in 2017 and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2016RIC1B1014774).

References

- [1] A. Iliopoulos, R. Shirzadeh, W. Weijtjens, P. Guillaume, D. V. Hemelrijck, and C. Devriendt, "A modal decomposition and expansion approach for prediction of dynamic responses on a monopile offshore wind turbine using a limited number of vibration sensors," *Mechanical Systems and Signal Processing*, vol. 68-69, pp. 84–104, 2016.
- [2] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas, "Protein classification with multiple algorithms," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 3746, pp. 448–456, 2005.
- [3] Y. Rao, "Contextual Sentiment Topic Model for Adaptive Social Emotion Classification," *IEEE Intelligent Systems*, vol. 31, no. 1, pp. 41–47, 2016.
- [4] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma, "Multi-label learning with millions of labels," in *Proceedings of the the 22nd international conference*, pp. 13–24, Rio de Janeiro, Brazil, May 2013.
- [5] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth, "Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary," in *Computer Vision — ECCV*

- 2002, vol. 2353 of *Lecture Notes in Computer Science*, pp. 97–112, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
- [6] H. Ghasemzadeh, N. Amini, R. Saeedi, and M. Sarrafzadeh, “Power-aware computing in wearable sensor networks: An optimal feature selection,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 800–812, 2015.
 - [7] B. Nushi, A. Singla, A. Krause, and D. Kossmann, “Learning and feature selection under budget constraints in crowdsourcing,” in *Proceedings of the 4th AAAI Conf. Human Computation and Crowdsourcing*, pp. 159–168, Austin, USA, October 2016.
 - [8] H. Yang, R. Fujimaki, Y. Kusumura, and J. Liu, “Online feature selection: A limited-memory substitution algorithm and its asynchronous parallel variation,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*, pp. 1945–1954, USA, August 2016.
 - [9] H. Blume, B. Bischl, M. Botteck et al., “Huge music archives on mobile devices,” *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 24–39, 2011.
 - [10] P. Naula, A. Airola, T. Salakoski, and T. Pahikkala, “Multi-label learning under feature extraction budgets,” *Pattern Recognition Letters*, vol. 40, no. 1, pp. 56–65, 2014.
 - [11] J. Lee and D. W. Kim, “Memetic feature selection algorithm for multi-label classification,” *Information Sciences*, vol. 293, pp. 80–96, 2015.
 - [12] M.-L. Zhang, J. M. Peña, and V. Robles, “Feature selection for multi-label naive Bayes classification,” *Information Sciences*, vol. 179, no. 19, pp. 3218–3229, 2009.
 - [13] H. Yang, Z. Xu, M. R. Lyu, and I. King, “Budget constrained non-monotonic feature selection,” *Neural Networks*, vol. 71, pp. 214–224, 2015.
 - [14] F. Min and J. Xu, “Semi-greedy heuristics for feature selection with test cost constraints,” *Granular Computing*, vol. 1, no. 3, pp. 199–211, 2016.
 - [15] J. Wang, P. Zhao, S. C. H. Hoi, and R. Jin, “Online feature selection and its applications,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 698–710, 2014.
 - [16] J. Lee and D.-W. Kim, “SCLS: Multi-label feature selection based on scalable criterion for large label set,” *Pattern Recognition*, vol. 66, pp. 342–352, 2017.
 - [17] J. Lee, H. Lim, and D.-W. Kim, “Approximating mutual information for multi-label feature selection,” *IEEE Electronics Letters*, vol. 48, no. 15, pp. 929–930, 2012.
 - [18] H. Lim, J. Lee, and D.-W. Kim, “Multi-label learning using mathematical programming,” *IEICE Transaction on Information and Systems*, vol. E98D, no. 1, pp. 197–200, 2015.
 - [19] Y. Lin, Q. Hu, J. Liu, and J. Duan, “Multi-label feature selection based on max-dependency and min-redundancy,” *Neurocomputing*, vol. 168, pp. 92–103, 2015.
 - [20] G. Doquire and M. Verleysen, “Mutual information-based feature selection for multilabel classification,” *Neurocomputing*, vol. 122, pp. 148–155, 2013.
 - [21] D. Kong, C. Ding, H. Huang, and H. Zhao, “Multi-label ReliefF and F-statistic feature selections for image annotation,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 2352–2359, usa, June 2012.
 - [22] J. Lee and D.-W. Kim, “Fast multi-label feature selection based on information-theoretic feature ranking,” *Pattern Recognition*, vol. 48, no. 9, pp. 2761–2771, 2015.
 - [23] X. Kong and P. S. Yu, “GMLC: A multi-label feature selection framework for graph classification,” *Knowledge and Information Systems*, vol. 31, no. 2, pp. 281–305, 2012.
 - [24] J. Lee and D.-W. Kim, “Feature selection for multi-label classification using multivariate mutual information,” *Pattern Recognition Letters*, vol. 34, no. 3, pp. 349–357, 2013.
 - [25] N. Spolaór, E. A. Cherman, M. C. Monard, and H. D. Lee, “A comparison of multi-label feature selection methods using the problem transformation approach,” *Electronic Notes in Theoretical Computer Science*, vol. 292, pp. 135–151, 2013.
 - [26] Q. Gu, Z. Li, and J. Han, “Correlated multi-label feature selection,” in *Proceedings of the the 20th ACM international conference*, p. 1087, Glasgow, Scotland, UK, October 2011.
 - [27] B. Qian and I. Davidson, “Semi-supervised dimension reduction for multi-label classification,” in *Proceedings of the Proc. 24th AAAI Conf. Artificial Intelligence*, pp. 569–574, Atlanta, USA, Jul 2010.
 - [28] B. Xue, M. Zhang, W. N. Browne, and X. Yao, “A Survey on Evolutionary Computation Approaches to Feature Selection,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.
 - [29] Y. Zhang, D. Gong, Y. Hu, and W. Zhang, “Feature selection algorithm based on bare bones particle swarm optimization,” *Neurocomputing*, vol. 148, pp. 150–157, 2015.
 - [30] Y. Zhang, D.-W. Gong, and J. Cheng, “Multi-objective particle swarm optimization approach for cost-based feature selection in classification,” *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64–75, 2017.
 - [31] Y. Zhang, X.-F. Song, and D.-W. Gong, “A return-cost-based binary firefly algorithm for feature selection,” *Information Sciences*, vol. 418–419, pp. 561–574, 2017.
 - [32] J. Read, B. Pfahringer, and G. Holmes, “Multi-label Classification Using Ensembles of Pruned Sets,” in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM)*, pp. 995–1000, Pisa, Italy, December 2008.
 - [33] N. Spolaór, M. C. Monard, G. Tsoumakas, and H. D. Lee, “A systematic review of multi-label feature selection and a new method based on label construction,” *Neurocomputing*, vol. 180, pp. 3–15, 2016.
 - [34] Y. Sun, A. K. C. Wong, and M. S. Kamel, “Classification of imbalanced data: a review,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.
 - [35] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Random k-labelsets for multilabel classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.
 - [36] S. Ji and J. Ye, “Linear dimensionality reduction for multi-label classification,” in *Proceedings of the Proc. 21th Int. Joint Conf. Artificial Intelligence*, pp. 1077–1082, Pasadena, USA, Jul 2009.
 - [37] F. Nie, H. Huang, X. Cai, and C. H. Ding, “Efficient and robust feature selection via joint $l_{2,1}$ -norms minimization,” in *Advances in Neural Information Processing System*, pp. 1813–1821, MIT Press, 2010.
 - [38] J. Lee and D.-W. Kim, “Mutual Information-based multi-label feature selection using interaction information,” *Expert Systems with Applications*, vol. 42, no. 4, pp. 2013–2025, 2015.
 - [39] M.-L. Zhang and Z.-H. Zhou, “A review on multi-label learning algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.
 - [40] Z. Zhu, S. Jia, and Z. Ji, “Towards a memetic feature selection paradigm,” *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 41–53, 2010.
 - [41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE*

- Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [42] J. Yin, T. Tao, and J. Xu, “A Multi-label feature selection algorithm based on multi-objective optimization,” in *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2015*, Ireland, July 2015.
 - [43] Y. Zhang, D.-W. Gong, X.-Y. Sun, and Y.-N. Guo, “A PSO-based multi-objective multi-label feature selection method in classification,” *Scientific Reports*, vol. 7, no. 1, article no. 376, 2017.
 - [44] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, “Parameter Control in Evolutionary Algorithms: Trends and Challenges,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 167–187, 2015.
 - [45] M.-L. Zhang and L. Wu, “LIFT: multi-label learning with label-specific features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2015.
 - [46] Y. Yang and S. Gopal, “Multilabel classification with meta-level features in a learning-to-rank framework,” *Machine Learning*, vol. 88, no. 1-2, pp. 47–68, 2012.
 - [47] A. Cano, J. M. Luna, E. L. Gibaja, and S. Ventura, “LAIM discretization for multi-label data,” *Information Sciences*, vol. 330, pp. 370–384, 2016.
 - [48] F. van den Bergh and A. P. Engelbrecht, “A study of particle swarm optimization particle trajectories,” *Information Sciences*, vol. 176, no. 8, pp. 937–971, 2006.
 - [49] S. Arlot and A. Celisse, “A survey of cross-validation procedures for model selection,” *Statistics Surveys*, vol. 4, pp. 40–79, 2010.
 - [50] J. Lee, H. Kim, N.-R. Kim, and J.-H. Lee, “An approach for multi-label classification by directed acyclic graph with label correlation maximization,” *Information Sciences*, vol. 351, pp. 101–114, 2016.
 - [51] F. Wilcoxon, “Probability tables for individual comparisons by ranking methods,” *Biometrics - A Journal of the International Biometric Society*, vol. 3, pp. 119–122, 1947.

Research Article

Experimental Verification of Optimized Multiscroll Chaotic Oscillators Based on Irregular Saturated Functions

J. M. Muñoz-Pacheco ¹, D. K. Guevara-Flores,¹ O. G. Félix-Beltrán,¹ E. Tlelo-Cuautle ²,
J. E. Barradas-Guevara,³ and C. K. Volos⁴

¹Faculty of Electronics Sciences, Autonomous University of Puebla, 72570 Puebla, PUE, Mexico

²Department of Electronics, National Institute of Astrophysics, Optics and Electronics, 72840 Tonantzintla, PUE, Mexico

³Faculty of Physics and Mathematics Sciences, Autonomous University of Puebla, 72570 Puebla, PUE, Mexico

⁴Laboratory of Nonlinear Systems, Circuits & Complexity, Department of Physics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

Correspondence should be addressed to J. M. Muñoz-Pacheco; jesusm.pacheco@correo.buap.mx

Received 1 October 2017; Revised 25 December 2017; Accepted 30 January 2018; Published 6 March 2018

Academic Editor: Kevin Wong

Copyright © 2018 J. M. Muñoz-Pacheco 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.

Multiscroll chaotic attractors generated by irregular saturated nonlinear functions with optimized positive Lyapunov exponent are designed and implemented. The saturated nonlinear functions are designed in an irregular way by modifying their parameters such as slopes, delays between slopes, and breakpoints. Then, the positive Lyapunov exponent is optimized using the differential evolution algorithm to obtain chaotic attractors with 2 to 5 scrolls. We observed that the resulting chaotic attractors present more complex dynamics when different patterns of irregular saturated nonlinear functions are considered. After that, the optimized chaotic oscillators are physically implemented with an analog discrete circuit to validate the use of proposed irregular saturated functions. Experimental results are consistent with MATLAB™ and SPICE circuit simulator. Finally, the synchronization between optimized and nonoptimized chaotic oscillators is demonstrated.

1. Introduction

The chaotic behavior has attracted a lot of attention for scientific community due to extreme sensitivity to its initial conditions and the broadband nature of its chaotic signals [1–28]. Therefore, in the last years, literature is vast in papers oriented to study new chaotic systems [1–5], propose novel applications [6–9], increase the degree of chaos (hyperchaotic systems [8, 10, 11]), get fractional order chaotic systems [7, 12, 13], synchronize the chaotic behavior [14–18], optimize chaotic systems [19–22], and implement chaotic oscillators using electronic circuits [23–28]. In all these studies, chaos behavior is analyzed and verified by using different approaches, for example, frequency spectrum, Poincaré maps, bifurcation diagrams, Lyapunov exponents, and stability of equilibrium points. Among them, Lyapunov exponents provide a direct measure of the sensitive dependence

on initial conditions by quantifying the exponential rates at which neighboring orbits on an attractor diverge as the system evolves in time [29–31].

For an n -dimensional nonlinear system, if the system has at least one positive Lyapunov exponent (LE) and is purely deterministic, then it is chaotic. Indeed, a tool commonly used to determine the presence of chaos in several numerical and experimental results is to compute only the positive LE [31].

Besides, the positive LE can be very useful to determine the unpredictability grade of the chaotic oscillator because its magnitude specifies the maximum average exponential rate corresponding to divergence of trajectories on an attractor and thus the maximum amount of instability along any direction [29–31]. That is, a high value of the positive LE can be taken as an indication of a high degree of chaos in the dynamical system [19–22, 32–35].

For instance, in [32] the speed effects and leg amputations on the dynamic stability of running are analyzed by computing the largest LE. The results revealed that the value of the largest LE is positive not only for unaffected patients but also for the one-leg affected patients. However, the positive LE of embedded time-series data from the affected leg was higher than for the unaffected leg indicating a more rich dynamics. In [33] a numerical scheme based on the Lattice Boltzmann method for the flow in complex mixer geometries to compute trajectories of passive tracers for the quantification of chaotic mixing was reported. They reported a better efficiency in chaotic micromixers when the value of positive LE was higher. Moreover, chaotic systems with a high value of the positive LE have also been used to improve the performance of chaos-based applications, for example, in [34] it was demonstrated that a high value of the positive LE in an optimization algorithm based on a particle swarm implies that the particles are inclined to explore different regions and find better fitness values. Therefore, the particle swarm with just a little variation in the value of positive LE usually achieved a better performance, especially for multimodal functions. In [35] the efficiency of hybrid chaotic optimization algorithms was studied by revealing effects on the search speed as a function of chaotic sequences from different chaotic maps. It was found that the higher the magnitude of positive LE, the faster the search speed in whole optimization space. Accordingly, the efficiency of global optimization was directly proportional to the value of positive LE.

In this framework, optimized chaotic systems with a high value of the positive LE can be extremely suitable to enhance the existing chaos-based applications. In electronics, a great variety of multiscroll chaotic oscillators has been implemented with commercially available electronic devices, as well as with integrated circuits technology [5, 10, 23, 26–28, 36, 37]. However, those experimental realizations are not optimized to provide a high value of the positive Lyapunov exponent (PLE). Although some authors have already used optimization algorithms based on evolutionary computation to get optimized multiscrolls chaotic systems [19, 22], they were obtained by using piecewise-linear (PWL) functions in the form of saturated nonlinear functions (SNLF) with symmetry properties. In addition, the experimental verification of those approaches is lacking.

This paper is motivated by the aforementioned discussion. In that scenario, we design irregular SNLF to obtain multiscroll chaotic attractors with optimized values of the positive LE. Additionally, we also demonstrate its practical feasibility by the physical implementation of the resulting multiscroll chaotic oscillators. Two cases were considered to design the irregular SNLFs. The first one consists of changing the breakpoints of SNLF to get different slopes, whereas the second one modifies the delay between slopes in different sections of SNLF. In both cases, once the parameters of SNLF are defined, we apply the evolutionary algorithms reported in [19, 22] to find the optimal value for system's parameters which maximizes the magnitude of positive LE. As a result, multiscroll chaotic oscillators with a more complex dynamics

are generated. Experimental results for 2-, 3-, 4-, and 5-scroll chaotic attractors were obtained with the aim of an analog discrete circuit based on commercial operational amplifiers (OpAmps). Further, we show the synchronization of those optimized chaotic oscillators by using generalized Hamiltonian forms because they can enhance the synchronization and realization of secure communication systems, for instance.

The paper is organized as follows. Section 2 describes the multiscroll chaotic oscillator under study; Section 3 outlines the steps to obtain optimized values of positive LE as well as the electronic design. Sections 4 and 5 present the experimental confirmation of the proposed approach for the two cases: slopes varying and different delays between slopes, respectively. Section 6 demonstrates the synchronization between optimized and nonoptimized chaotic oscillators. Finally, conclusions are given in Section 7.

2. SNLF-Based Multiscroll Chaotic Oscillator

The case of study in this work is the chaotic oscillator described by

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= z, \\ \dot{z} &= -ax - by - cz + d_1 f(x; q),\end{aligned}\tag{1}$$

where (x, y, z) is the state variables, $f(x; q)$ is the SNLF, and (a, b, c, d_1) is the system's parameters. To maximize the value of positive LE requires varying the coefficients of chaotic oscillator, leading to a huge number of combinations. Herein, system's parameters (a, b, c, d_1) are varied within the range $[0.0001 \cdots 1.0000]$. So, we define four variables where each one can have $(2 \times 10^4)^4 = 16 \times 10^{16}$ possible combinations. This result justifies the application of heuristics like the ones already introduced in [19, 22]. For all cases analyzed in this work, the phase-space plots show the state variables x versus y .

The first step consists of manipulating $f(x; q)$ with the goal of incrementing the complexity of multiscroll chaotic oscillator. To generate 2 scrolls, the SNLF description is given by

$$f(x; q) = \begin{cases} k, & \text{if } x > q(2), \\ \frac{k}{q}, & \text{if } q(2) > x > q(1), \\ -k, & \text{if } x < q(1), \end{cases}\tag{2}$$

where k represents the value of saturated regions, $kv = k/q$ is the slope between two saturated regions, q is a break point connecting a saturated region with a slope, and h is the delay, as shown in Figure 1. In general, the number of scrolls to be generated equals the number of saturated regions k . Therefore, (2) can be augmented to generate n -scrolls as shown in [38]. It means that by augmenting segments in a symmetric way as shown in Figures 1(a) and 1(b), respectively, even and odd number of scrolls are generated. However, in

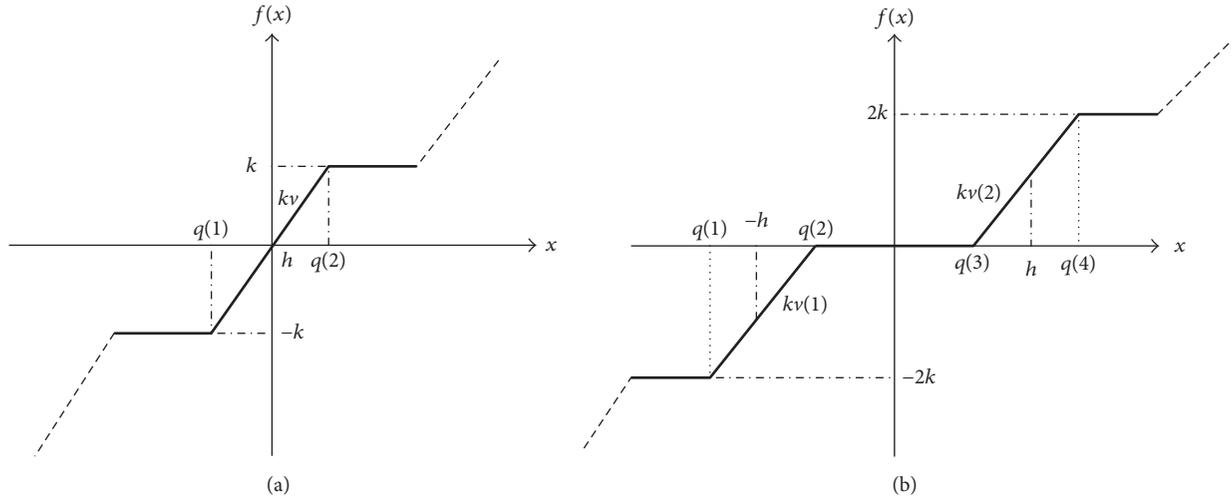


FIGURE 1: Description of a typical SNLF function for generating (a) even and (b) odd scrolls.

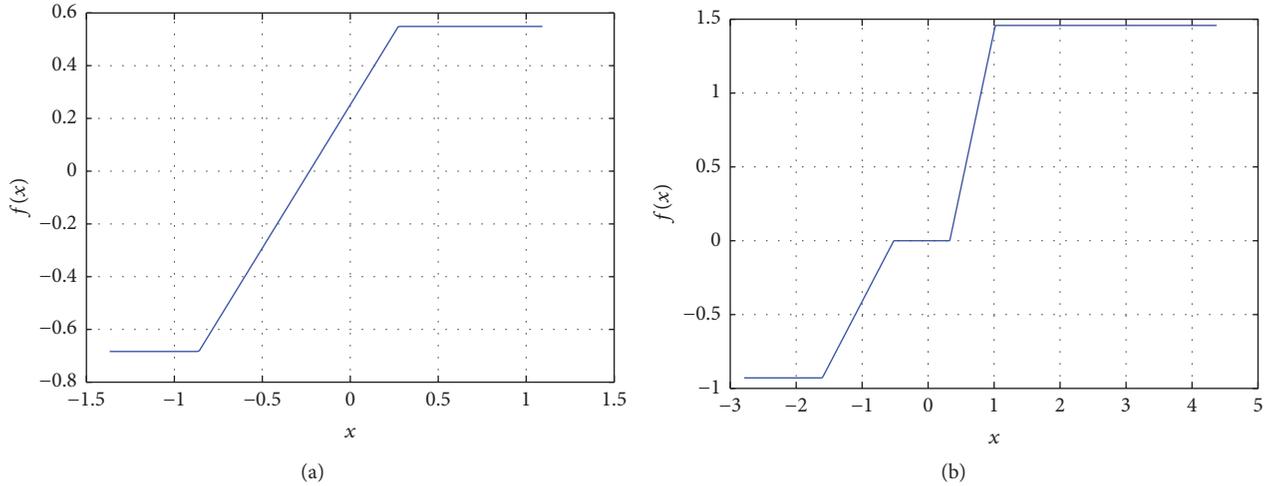


FIGURE 2: Examples of irregular SNLFs with (a) 2 and (b) 3 saturated levels.

this work we show how to use irregular SNLF functions, that is, nonsymmetric, to obtain multiscrolls as the ones shown in Figure 2. The irregular SNLF functions are herein designed using different values for breakpoints $q(l)$ with $l = 1, 2, 3, \dots, j$ where j is the number of breakpoints, slopes kv , and saturated levels k .

3. Methodology to Maximize the Positive LE of Multiscrolls Oscillators and Its Circuit Design

The solution to (1) by using symmetric SNLF described by (2) has been performed by applying evolutionary algorithms in [19, 22], where the positive LE was optimized. By applying the differential evolution (DE) algorithm already introduced in [22], we obtained several feasible solutions for the coefficients a, b, c , and d_1 , which provide higher values of the positive

LE as the number of scrolls is incremented. In all cases, the value of positive LE computed by DE is higher than when using traditional values (TV) for system's parameters as $TV = \{a = b = c = d_1 = 0.7\}$ [19]. Details of applying DE algorithm to optimize the positive LE can be found in [22].

On the other hand, the SNLF can be implemented as shown in Figure 3, where the number of saturated levels (SL) determines the number of OpAmps to be used considering $\#OpAmp = \#SL - 1$, as shown in [23]. In that reference the whole electronic realization of (1) is given by using OpAmps. That realization is redrawn in Figure 4, where the SNLF function is embedded in block PWL, and system's parameters a, b, c , and d_1 are implemented by the ratio of resistors R_{fa}/R_{ia} , R_{fb}/R_{ib} , R_{fc}/R_{ic} , and R_{fd}/R_{id} , respectively.

Otherwise, this article shows how realizing irregular SNLFs by modifying Figure 3, such as asymmetric variation of the slopes, modifying the breakpoints $q(l)$, $l = 1, \dots, m$, and asymmetric variation of the saturated levels k .

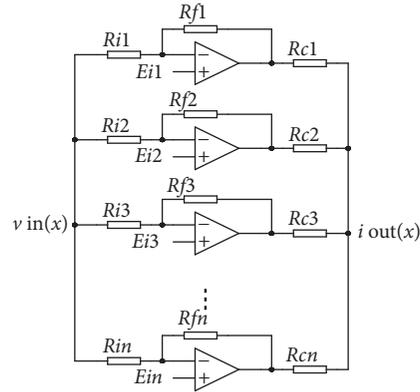


FIGURE 3: Realization of SNLFs using OpAmps represented in Figure 4 as PWL block.

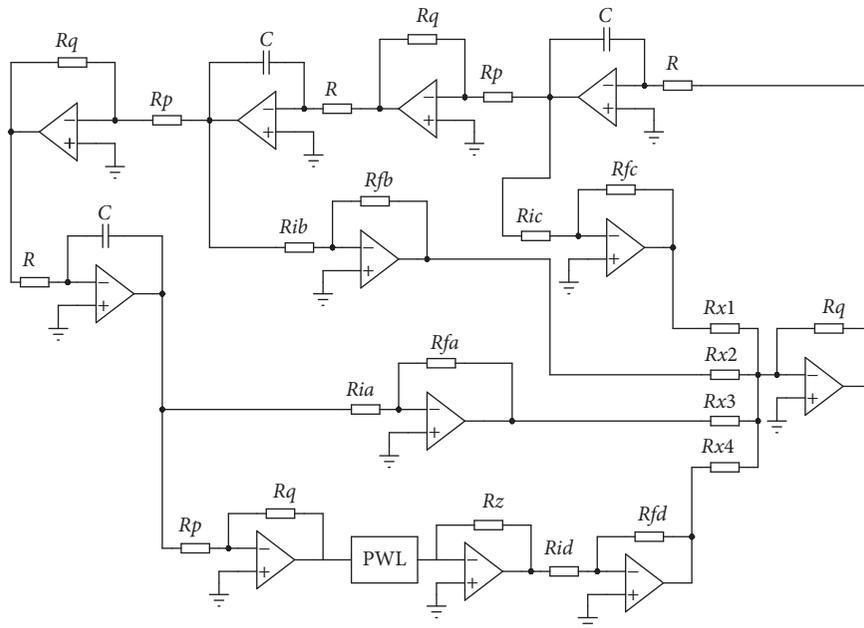


FIGURE 4: Realization of (1) using OpAmps.

The experiments are performed considering the following steps:

- (1) Modify the SNLFs to obtain an irregular one using MATLAB.
- (2) Optimize the value of positive LE by computing the optimal values for a, b, c , and d_1 with the DE algorithm given in [22].
- (3) Implement physically the SNLFs with OpAmps.
- (4) Validate experimentally the optimized chaotic oscillator by generating the required number of scrolls.

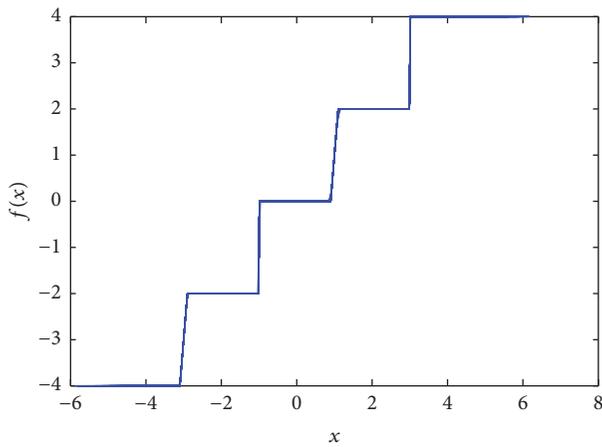
For instance, the MATLAB, SPICE, and experimental results of a 5-scrolls chaotic attractor obtained with an irregular SNLF are shown in Figure 5. More details are given in the following sections.

4. Experimental Results by Varying the Slopes of SNLFs

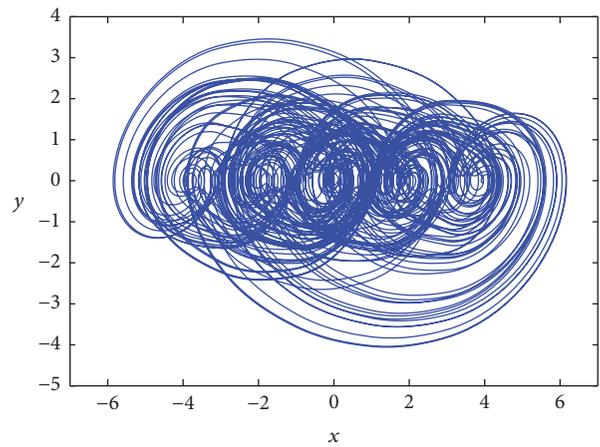
The slope in SNLF is described herein by k_v , as shown in Figure 1, which is modified asymmetrically to generate irregular SNLFs for obtaining 3, 4, and 5 scrolls. The first step consists of determining the minimum and maximum slopes values using MATLAB, so that the results to attain from 2 to 7 scrolls are given in Table 1. To generate more scrolls, the minimum and maximum values are the same as for 6 and 7 scrolls. Next, the values of circuit elements to get different slopes in SNLF are found. This is done by using commercially available OpAmps in Figure 3. Table 2 lists the calculated values to obtain slopes of 5, 10, 20, 30, 40, 50, and 100. In all cases it is assumed a saturation voltage $V_{\text{sat}} = 15 \text{ V}$ for the OpAmps. Using Table 2, one can combine

TABLE 1: Values of the slopes computed by MATLAB.

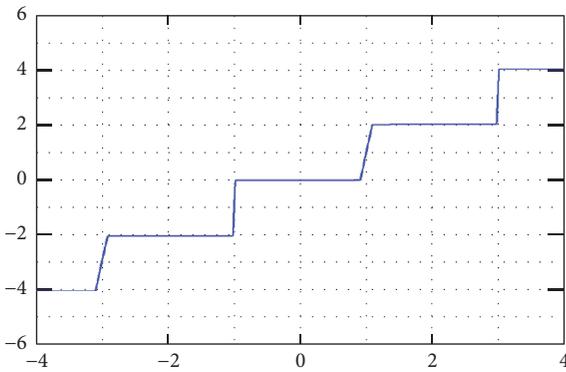
Scrolls	Minimum slope k_v	Maximum slope k_v
2	2	∞
3	4	∞
4	5	∞
5	7	∞
6	10	∞
7	10	∞



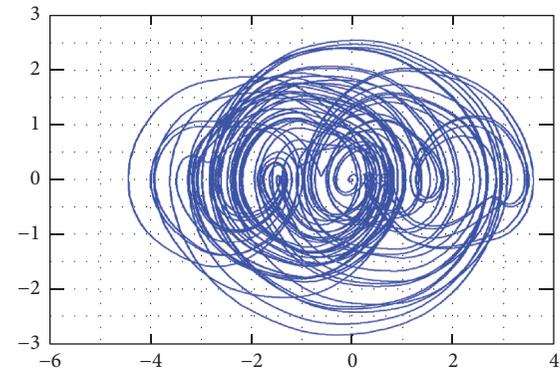
(a)



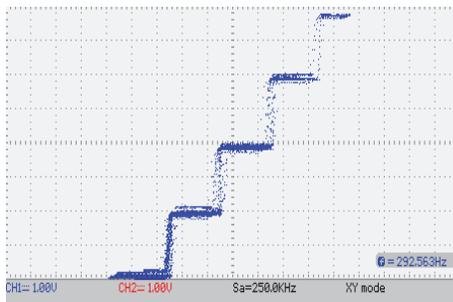
(b)



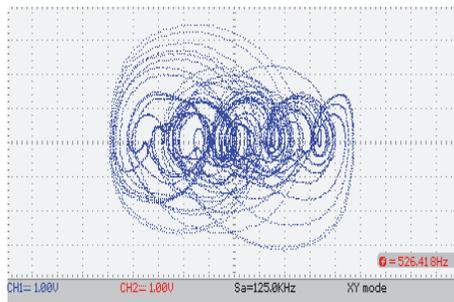
(c)



(d)



(e)



(f)

FIGURE 5: 5-scroll using irregular SNLF for parameters given in Table 2: irregular SNLF and chaotic attractor with (a), (b) MATLAB, (c), (d) SPICE, and (e), (f) experimental realization.

TABLE 2: Circuit element values for implementing different slopes using Figure 3.

Slope value	R_i (k Ω)	R_f (k Ω)	R_c (k Ω)
5	1.0	75	150
10	1.0	150	150
20	1.0	300	150
30	1.0	450	150
40	1.0	600	150
50	1.0	750	150
100	1.0	1500	150

TABLE 3: Combinations of different slopes for realizing irregular SNLFs generating 3 to 5 scrolls.

Scrolls	Case		
	1	2	3
3	[5, 10]	[10, 20]	[20, 50]
4	[10, 10, 5]	[50, 10, 20]	[30, 50, 70]
5	[10, 10, 20, 20]	[10, 30, 30, 10]	[10, 100, 10, 100]

TABLE 4: Positive LEs for the optimized multiscrolls chaotic oscillators with different slopes kv as the cases listed in Table 3.

Case	Scroll number	Positive LE without optimization	a	b	c	d_1	Positive LE with optimization and irregular SNLF
1	3	0.2578	1.000	0.751	0.394	0.981	0.5555
	4	0.2890	1.000	0.709	0.290	1.000	0.6327
	5	0.3279	1.000	0.781	0.170	1.000	0.6580
2	3	0.2737	0.997	0.783	0.395	0.988	0.5608
	4	0.3121	1.000	0.745	0.304	1.000	0.6470
	5	0.2763	1.000	0.658	0.262	0.999	0.6823
3	3	0.2651	1.000	1.000	0.270	0.998	0.5656
	4	0.2760	1.000	0.927	0.189	1.000	0.6667
	5	0.3016	1.000	0.680	0.262	1.000	0.7112

different slopes to realize irregular SNLFs. In this manner, Table 3 shows the combinations of slopes to generate 3, 4, and 5 scrolls. In the experiments, those values have been selected randomly from Table 2. Each combination of slopes is used to implement the irregular SNLF described by (2) and then to implement the dynamical system described in (1). All cases in Table 3 for generating 3-, 4-, and 5-scroll chaotic attractors with optimized positive LE are described below.

4.1. Optimized 3-Scrolls Chaotic Oscillator with Different Slopes. Using the irregular SNLF with the slopes listed in Table 3 and setting traditional values of $a = b = c = d_1 = 0.7$, the positive LE is listed in Table 4. This is positive for the three cases indicating chaotic behavior. Afterwards, by applying DE algorithm [22] the optimized positive LE provides new coefficient values that are also listed in Table 4. As supposed the value of positive LE is higher than those values without optimization. The optimization of positive LE applying [22] was executed using a population of 50 individuals and 50 iterations.

According to case 1 from Table 3 and from Figure 1, the generation of 3 scrolls with optimized positive LE implies

slopes being [5, 10], which are located in the third and first quadrants, respectively. Cases 2 and 3 require the slopes being [10, 20] and [20, 50], respectively. The circuit element values for realizing these slopes were already listed in Table 2. Figure 6 shows the experimental results for the irregular SNLF for each case and the associated attractor showing 3 scrolls with optimized positive LE.

4.2. Optimized 4-Scroll Chaotic Oscillator with Different Slopes. For generating 4 scrolls with optimized positive LE, the values of the slopes are also listed in Table 3. Again, by setting traditional values $a = b = c = d_1 = 0.7$, positive LE is listed in Table 4. As a result, it is more positive for the three cases than those values computed for 3 scrolls in Table 4. This confirms that positive LE increases by augmenting the number of scrolls.

The optimized positive LE provides new values for system's parameters that are listed in Table 4. As can be appreciated, the optimized positive LE is also higher for each case than for 3 scrolls. The optimization for positive LE applying [22] was executed using a population of 50 individuals and 50 iterations. The irregular SNLFs are realized

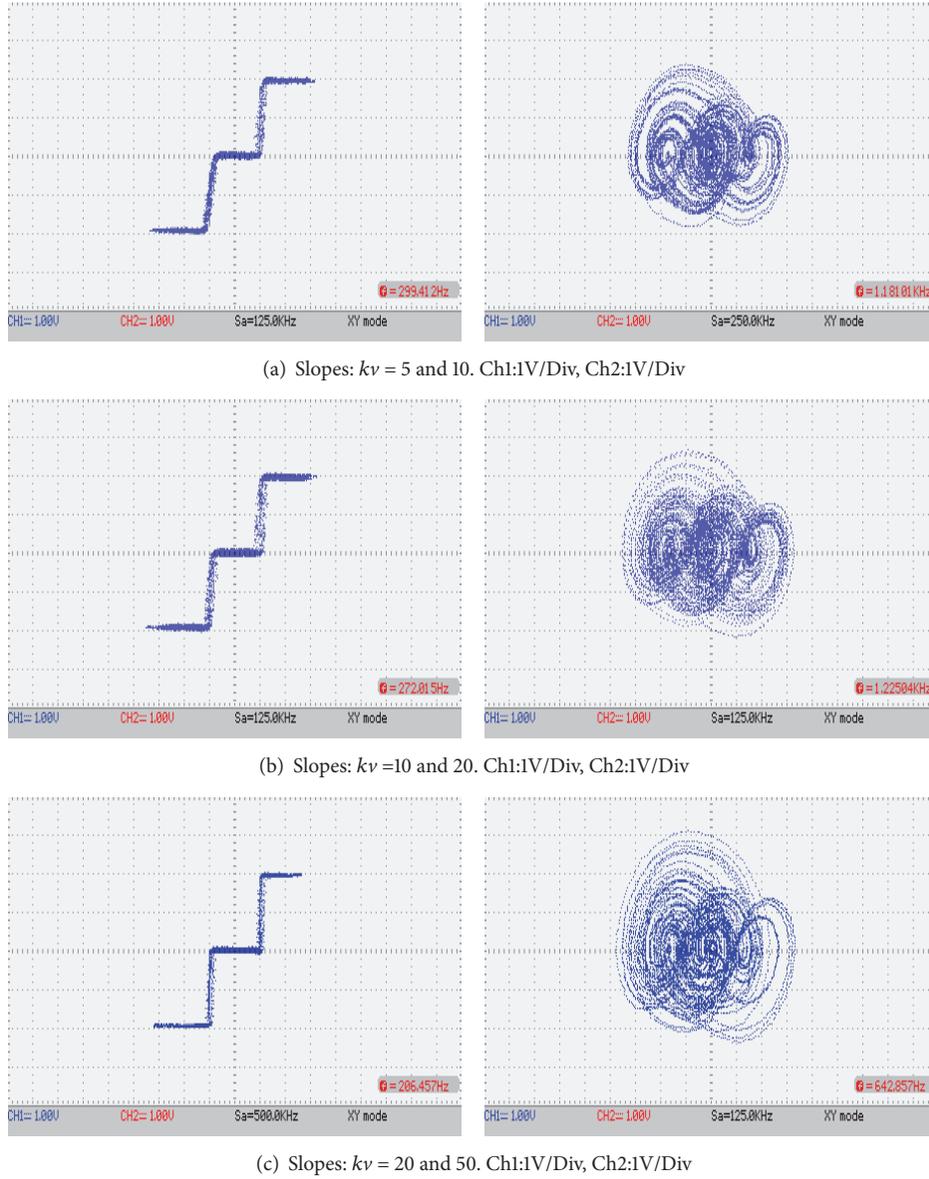


FIGURE 6: Experimental verification of optimized 3 scrolls using irregular SNLFs with different slopes given in Table 3.

again using the slopes listed in Table 3 and Figure 1. Again, the circuit element values for realizing the three slopes are taken from Table 2. Figure 7 shows the experimental results for the irregular SNLF and the associated chaotic attractor showing 4 scrolls. As one can infer, according to Table 1, the first case for 4 scrolls uses the minimum slope value of 5, which is also appreciated in the irregular SNLF shown in Figure 7(a).

4.3. Optimized 5-Scroll Chaotic Oscillator with Different Slopes. Generating 5 scrolls with optimized positive LE implies using four slopes, which are listed in Table 3. For those cases and by setting traditional values of $a = b = c = d_1 = 0.7$, the positive LE is listed in Table 4, where their magnitudes are slightly more higher than those ones

for 3 and 4 scrolls in Table 4. Again, positive LE increases by augmenting the number of scrolls from 4 to 5.

The optimized positive LE is listed in Table 4, which is also higher than for generating 4 scrolls. The irregular SNLFs are realized again using the slopes listed in Table 3 and Figure 1. Four slopes are used because the SNLF shown in Figure 1(b) is increased. Once again, the circuit element values are taken from Table 2. Figure 8 shows the experimental results for the irregular SNLF and the associated chaotic attractor showing 5 scrolls.

In this section the irregular SNLFs was realized by modifying the values of slopes k_v . The experiments confirmed the generation of 3, 4, and 5 scrolls that have an optimized positive LE, which was computed by applying DE algorithm in [22].

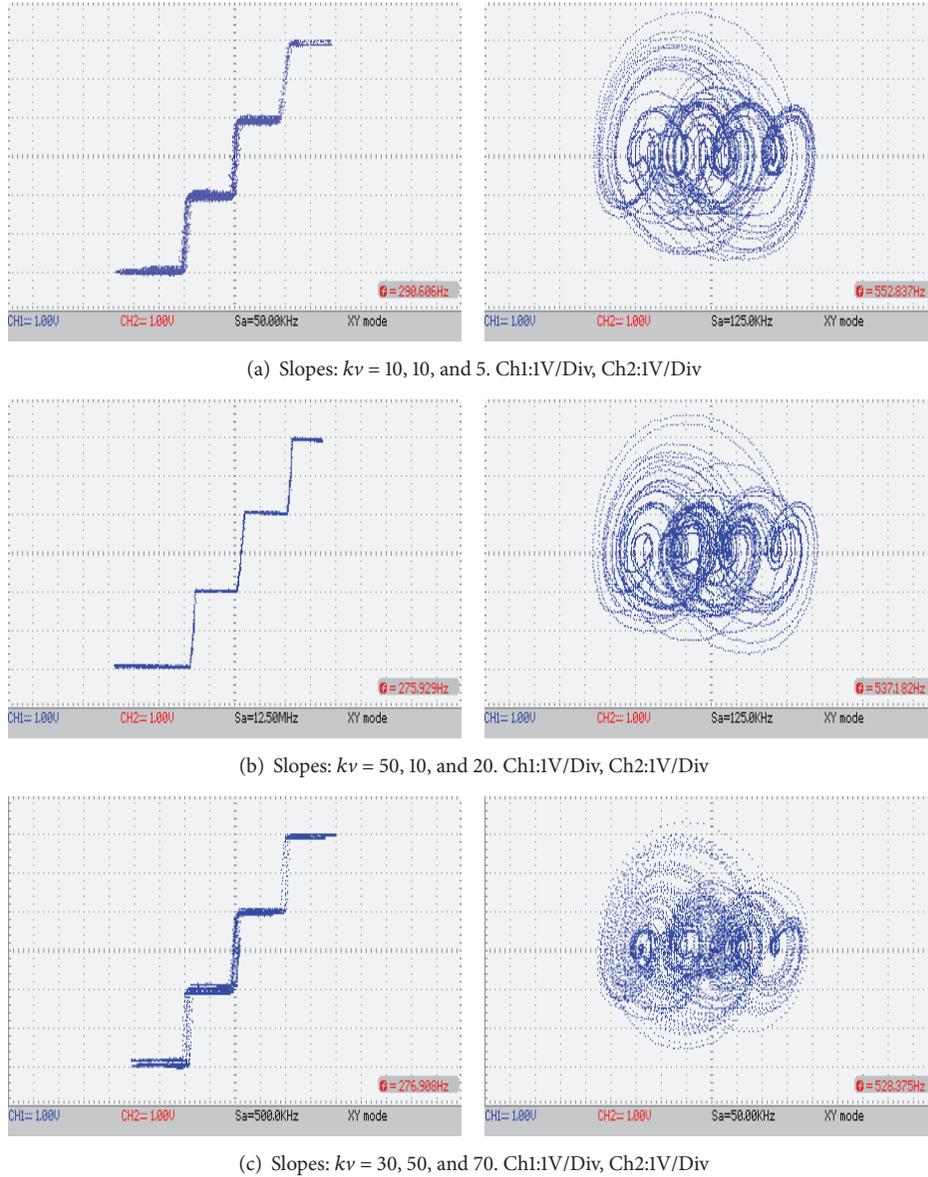


FIGURE 7: Experimental verification of optimized 4 scrolls using irregular SNLFs with different slopes given in Table 3.

5. Experimental Results by Varying the Delay of Slopes in SNLFs

This section shows the experimental verification of optimized multiscroll chaotic oscillators by varying asymmetrically the delay h (see Figure 1) of slopes kv , that is, the distance separating the center of slope with respect to the horizontal axis. Contrary to the previous section, the value of slopes kv is kept as a constant for all cases, for example, $kv = 10$. Also, the saturation levels k are the same.

The physical realization of irregular SNLFs using commercially available OpAmps requires multiple voltage dividers to get the required voltages E_i in Figure 3. Table 5 lists the values of h for each slope, according to the number of scrolls to be generated.

5.1. Optimized 2-Scroll Chaotic Oscillator with Different Delays. Table 6 lists the positive LEs of three cases from Table 5 by setting traditional values of $a = b = c = d_1 = 0.7$. In this case, only case 1 shows chaotic regime because cases 2 and 3 do not have a positive LE. However, after applying the optimization algorithm from [22], all cases have a positive LE, as listed in Table 6.

The experimental realization is performed by using system's parameters listed in Table 6 and the irregular SNLF with delays h listed in Table 5. In case 1, $h = -0.5$ V for E_i in Figure 3, so that the saturation region k on the right is wider than the left one. This gives as a result a larger scroll in the region of larger saturation width, as shown in Figure 9(a). The center of slope kv defined by delay h is the connecting point of neighbouring scrolls, that is, at -0.5 V. The other

TABLE 5: Proposed values for modifying the delay h .

Scrolls	Case 1	Case 2	Case 3
2	$[-0.5]$	$[1]$	$[1.5]$
3	$[-1, 1.5]$	$[-2, 1]$	$[-1.5, 2]$
4	$[-2, 0.5, 2]$	$[-2.5, 0, 1.5]$	$[-1.5, 0, 2]$
5	$[-3, -1, 1.3, 2.5]$	$[-3, -1.5, 0.5, 2.5]$	$[-2.5, -1, 1, 2.5]$

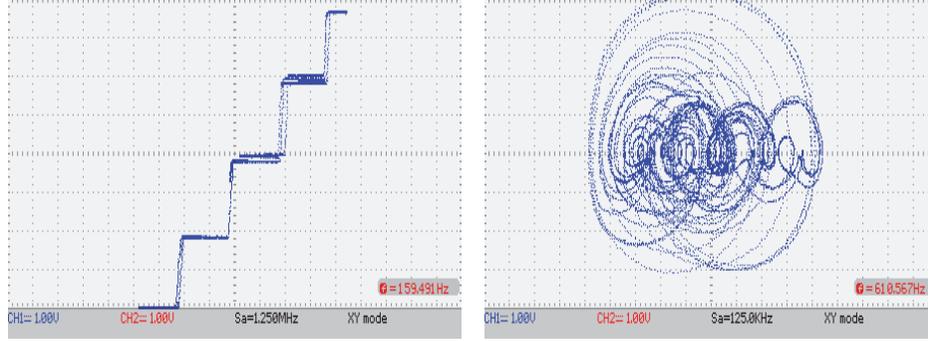
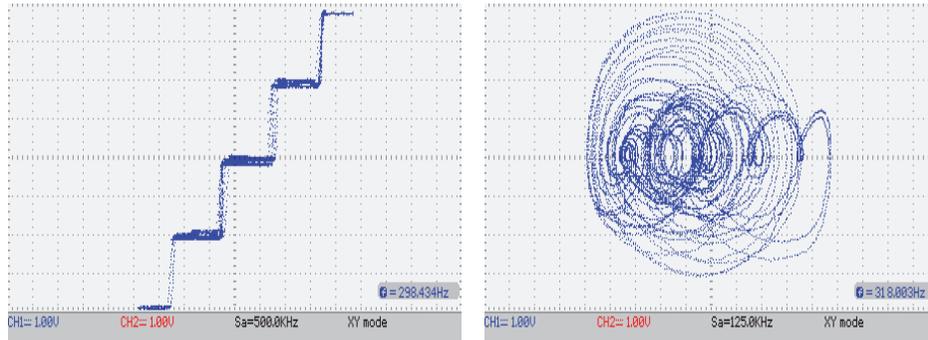
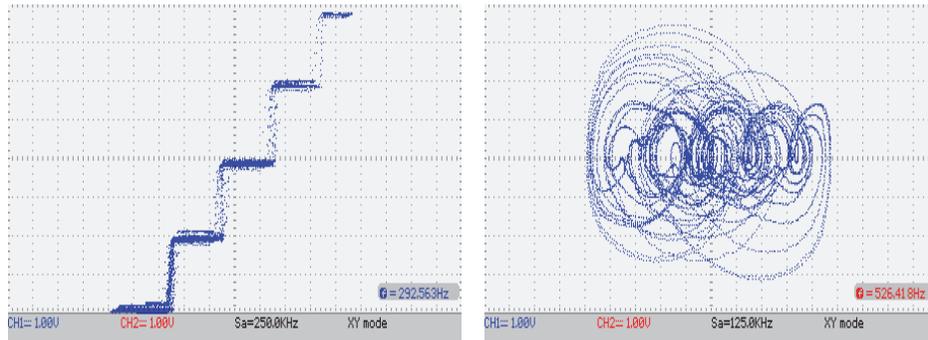
(a) Slopes: $kv = 10, 10, 20,$ and 20 . Ch1:1V/Div, Ch2:1V/Div(b) Slopes: $kv = 10, 30, 30,$ and 10 . Ch1:1V/Div, Ch2:1V/Div(c) Slopes: $kv = 10, 100, 10,$ and 100 . Ch1:1V/Div, Ch2:1V/Div

FIGURE 8: Experimental verification of optimized 5 scrolls using irregular SNLFs with different slopes given in Table 3.

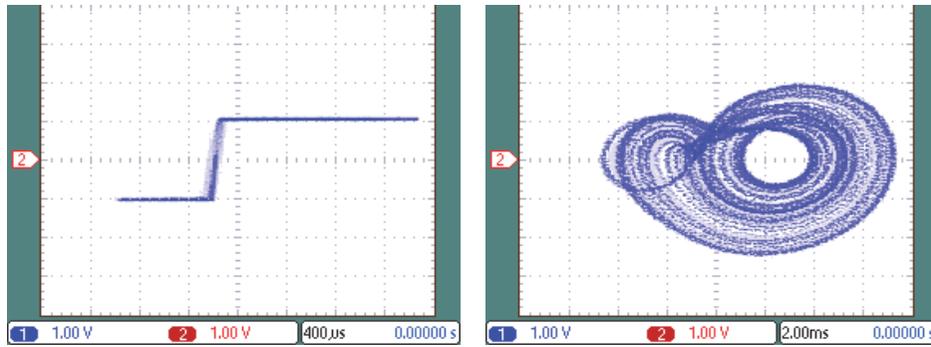
two cases are shown in Figures 9(b) and 9(c), respectively.

5.2. *Optimized 3-Scroll Chaotic Oscillator with Different Delays.* Table 6 lists the positive LE for generating 3 scrolls and by setting traditional values of $a = b = c = d_1 = 0.7$.

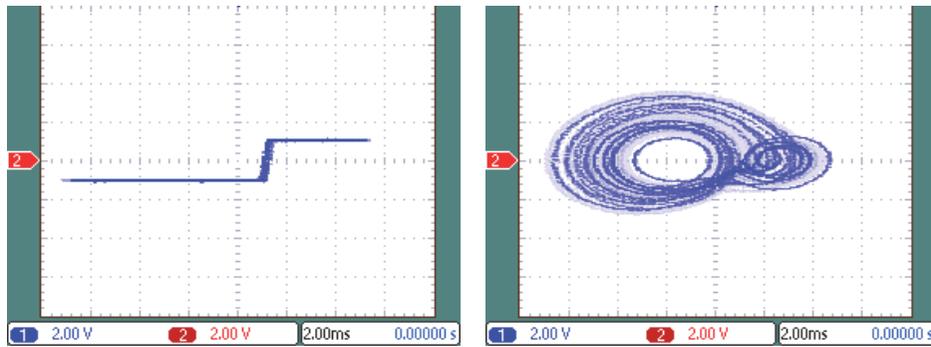
All cases have a positive LE. After applying the optimization algorithm from [22], the value of positive LE is increased as listed in Table 6. Again, as for the previous section, the chaotic complexity is being increased by augmenting the number of scrolls. From Table 5, the three cases for obtaining 3 scrolls and by varying the delay h are shown in Figure 10.

TABLE 6: Positive LEs for the optimized multiscrolls chaotic oscillators with different delays h as the cases listed in Table 5.

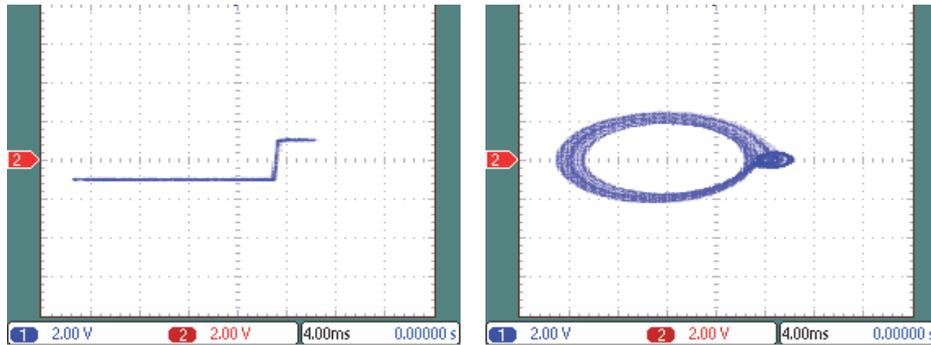
Case	Scroll number	Positive LE without optimization	a	b	c	d_1	Positive LE with optimization and irregular SNLF
1	2	0.0263	0.495	0.620	0.428	1.000	0.2671
	3	0.2166	0.747	0.618	0.490	0.722	0.3900
	4	0.2283	1.000	0.642	0.514	1.000	0.4318
	5	0.2865	0.851	0.693	0.292	0.731	0.4143
2	2	-0.0182	0.628	0.486	1.000	1.000	0.3385
	3	0.0250	0.625	0.585	0.367	0.997	0.4144
	4	-0.0324	0.998	0.735	0.504	0.774	0.4612
	5	0.0259	0.851	0.693	0.292	0.731	0.4143
3	2	0	0.553	0.919	0.607	1.000	0.1688
	3	0.2221	1.000	0.853	0.546	1.000	0.4238
	4	0.0261	0.975	0.903	0.319	1.000	0.4466
	5	0.0276	0.888	0.624	0.396	0.865	0.3725



(a) Case 1: $h = -0.5$. Ch1:1V/Div, Ch2:1V/Div



(b) Case 2: $h = 1$. Ch1:2V/Div, Ch2:2V/Div



(c) Case 3: $h = 1.5$. Ch1:2V/Div, Ch2:2V/Div

FIGURE 9: Experimental verification of optimized 2 scrolls using irregular SNLFs with different delays given in Table 5.

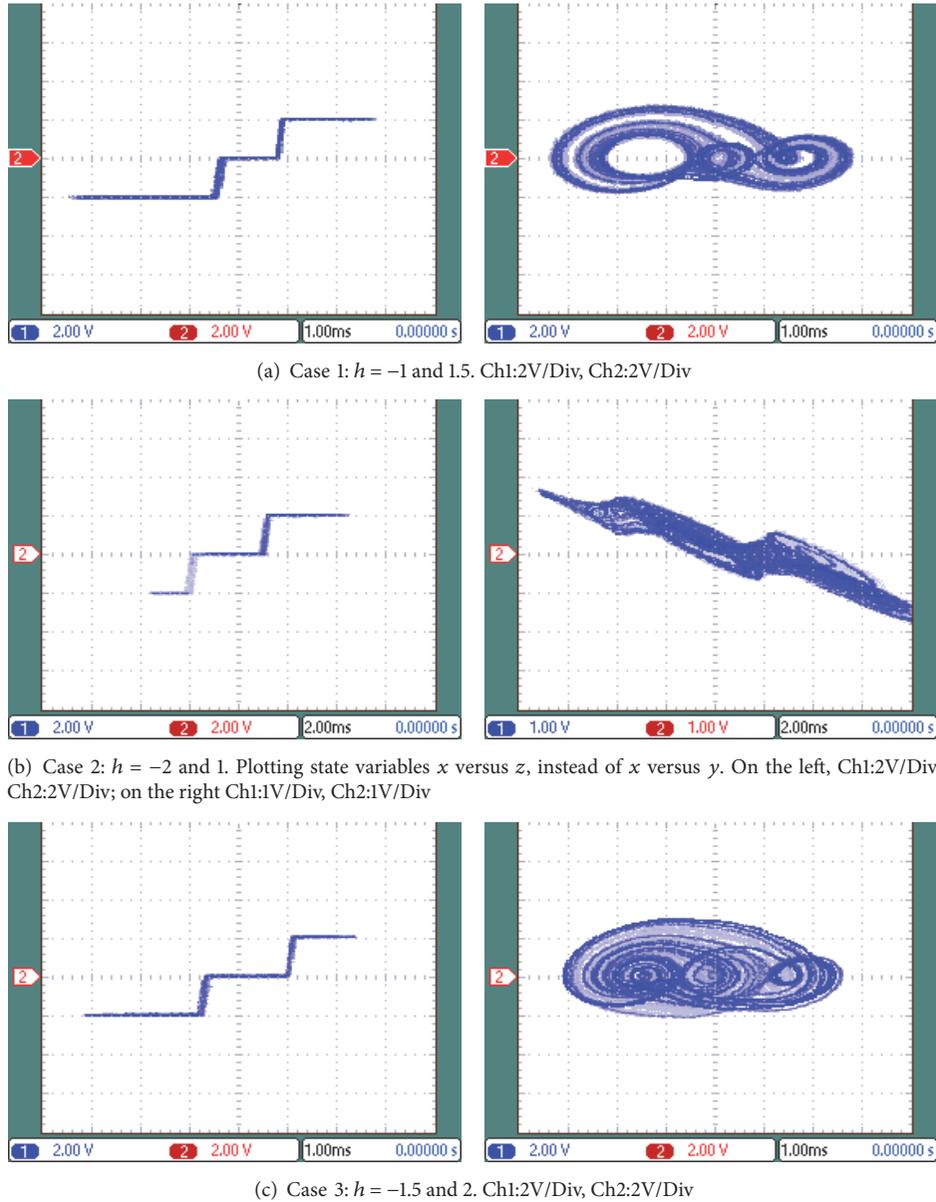


FIGURE 10: Experimental verification of optimized 3 scrolls using irregular SNLFs with different delays given in Table 5.

5.3. Optimized 4-Scroll Chaotic Oscillator with Different Delays. The three cases for getting 4 scrolls and by setting traditional values of $a = b = c = d_1 = 0.7$ have the positive LE listed in Table 6. Not all the cases have a positive LE; however, after applying the optimization DE algorithm from [22], all cases have a greater positive LE than for 3 scrolls, as listed in Table 6, thus confirming again that the chaotic complexity is being increased by augmenting the number of scrolls. From Table 5, the three cases for obtaining 4 scrolls and by varying the delay h are shown in Figure 11.

5.4. Optimized 5-Scroll Chaotic Oscillator with Different Delays. Finally, the three cases for getting 5 scrolls and by setting traditional values of $a = b = c = d_1 = 0.7$ have

the positive LE listed in Table 6. All the cases have a positive LE, and they are optimized by applying the optimization DE algorithm [22], where only case 3 has a positive LE greater than for generating 4 scrolls, as listed in Table 6. Anyway, the chaotic complexity is being increased by augmenting the number of scrolls. From Table 5, the three cases for attaining 5 scrolls and by varying the delay h are shown in Figure 12.

6. Synchronization between Optimized and Nonoptimized Multiscroll Chaotic Attractors

In this section the synchronization between optimized and nonoptimized multiscroll chaotic attractors is shown by

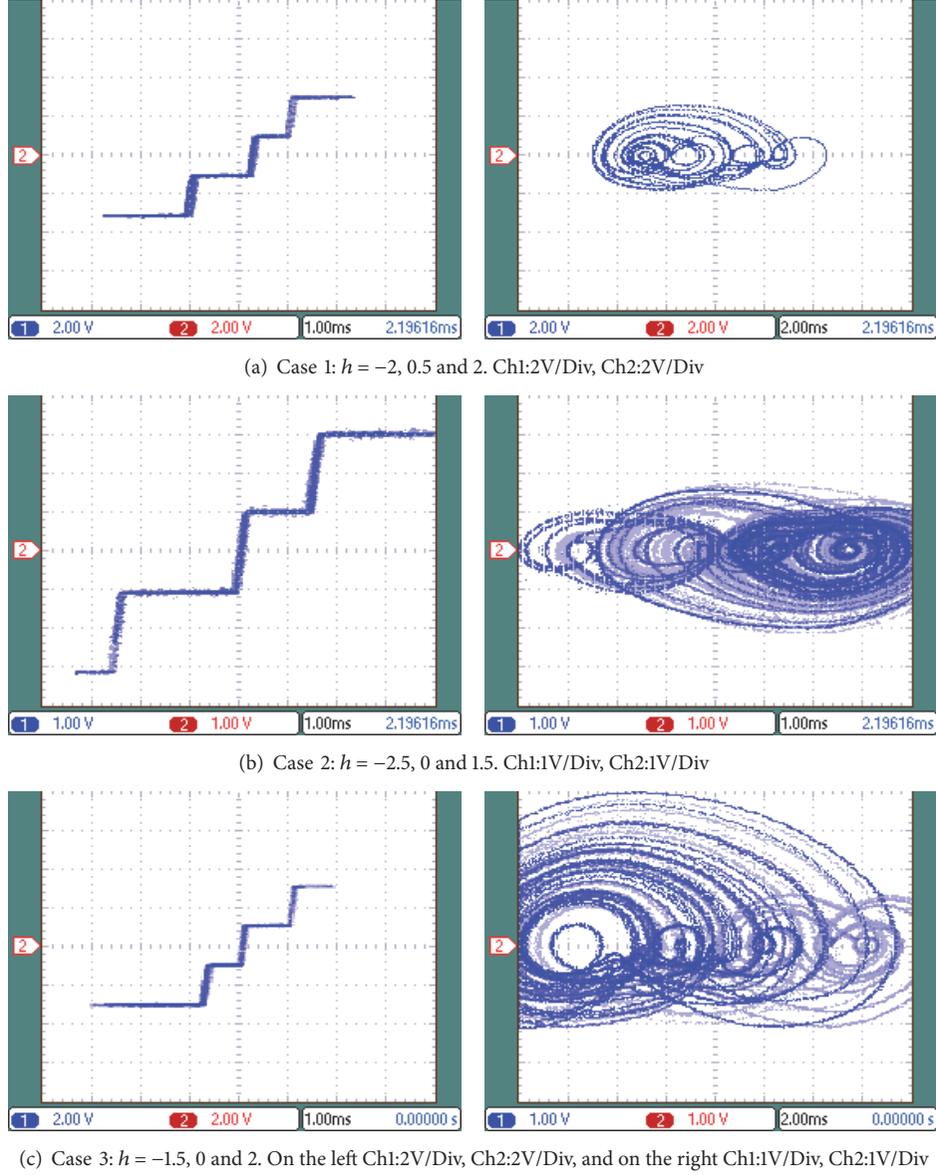


FIGURE 11: Experimental verification of optimized 4 scrolls using irregular SNLFs with different delays given in Table 5.

applying the proposed approach given in [39], that is, generalized Hamiltonian forms. Besides, it is demonstrated that, by using an optimized 4-scroll chaotic oscillator as master system, the slave system that is not optimized can show a similar behavior. This is an important result since it is not necessary to optimize each chaotic oscillator independently.

Then, let us take the multiscroll chaotic system (1) as the master system as follows:

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= z, \\ \dot{z} &= -ax - by - cz + d_1 f(x; q). \end{aligned} \quad (3)$$

By using a Hamilton energy function defined by $H(x) = 1/2[ax^2 + by^2 + z^2]$, we can obtain the slave system defined by

$$\begin{aligned} \dot{\hat{x}} &= \hat{y} + k_1(x - \hat{x}), \\ \dot{\hat{y}} &= \hat{z} + k_2(x - \hat{x}), \\ \dot{\hat{z}} &= -a\hat{x} - b\hat{y} - c\hat{z} + d_1 \hat{f}(x; \hat{q}) + k_3(x - \hat{x}), \end{aligned} \quad (4)$$

which synchronizes with master system (4) when the synchronization error $x - \hat{x}$ tends to zero by selecting appropriate synchronization gains k_1, k_2, k_3 . The synchronization gains are selected as explained in detail in [39].

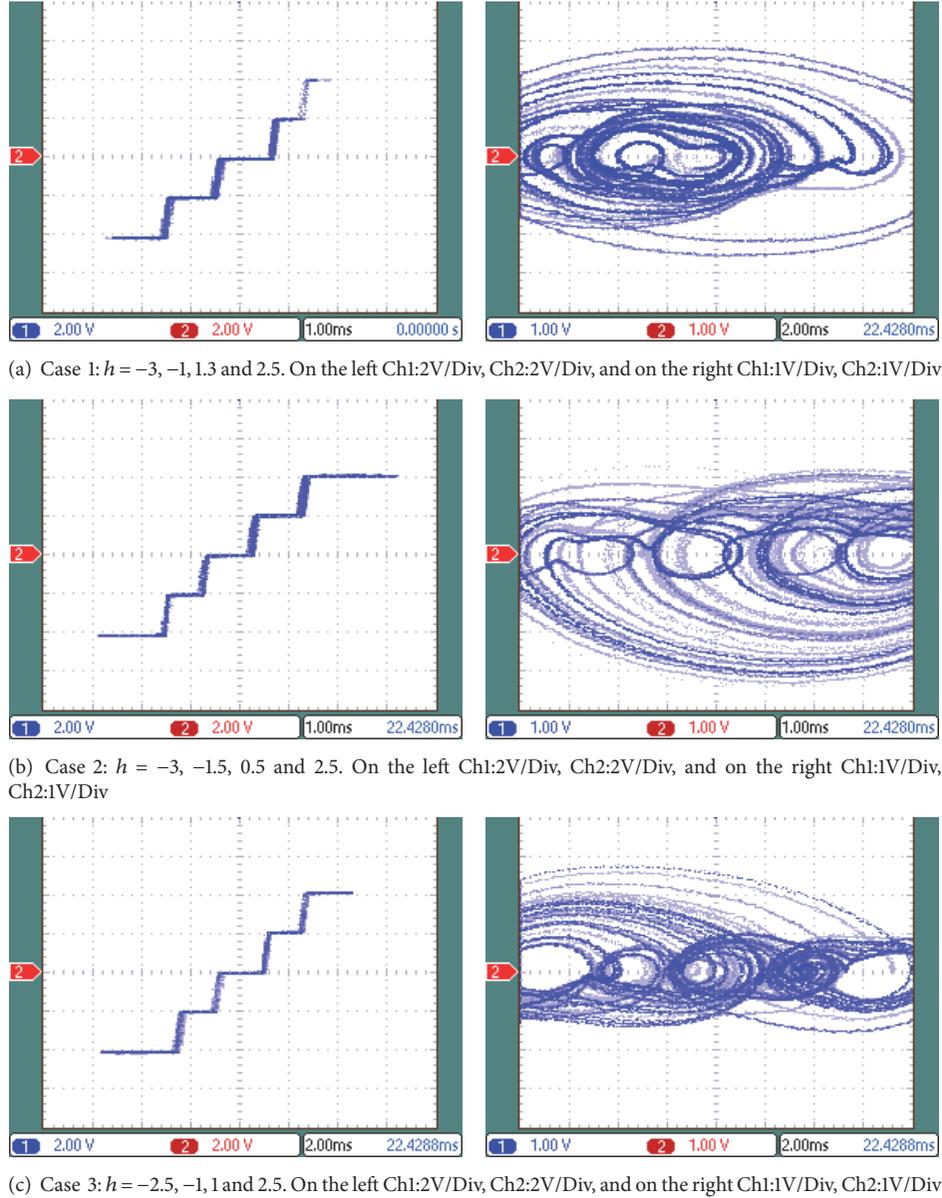
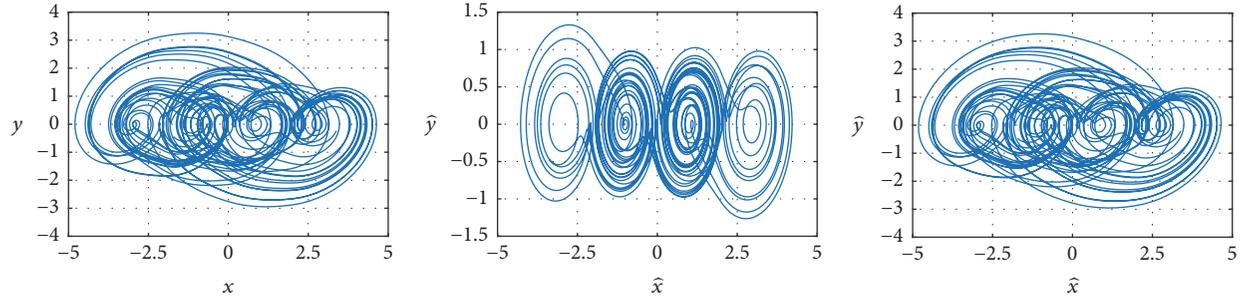


FIGURE 12: Experimental verification of optimized 5 scrolls using irregular SNLFs with different delays given in Table 5.

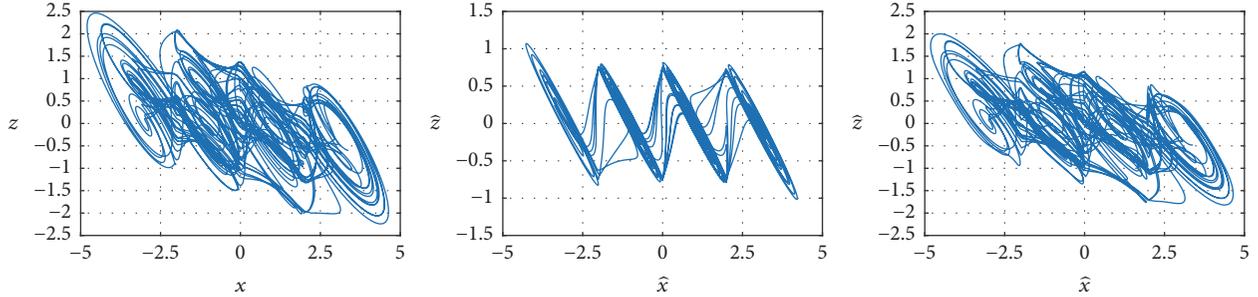
6.1. Synchronization of Nonoptimized 4-Scroll with Case 1 in Table 4: Different Slopes. By using different slopes for SNLF (irregular) as given in case 1, Table 4, the resulting 4-scroll chaotic attractor can be synchronized with another 4-scroll chaotic attractor that preserves traditional values for system's parameters as well as a regular SNLF. For master system, we use the slopes = 10, 10, 5 (case 1, Table 4) for irregular SNLF, system's parameters $a = 1.0$, $b = 0.709$, $c = 0.290$, $d_1 = 1.0$, synchronization gains $k_1 = 4$, $k_2 = 20$, $k_3 = 45$, and initial conditions $x(0) = 1$, $y(0) = 1$, $z(0) = 0.01$. On the other hand, for slave system traditional values $a = 0.7$, $b = 0.7$, $c = 0.7$, $d_1 = 0.7$ are considered; also a symmetrical SNLF with slopes = 200 are used, and $\hat{x}(0) = 0.1$, $\hat{y}(0) = 0.1$, $\hat{z}(0) = 0.1$. Figure 13 shows the simulation results of synchronization. In Figure 13(a) the phase portraits for master system and slave

system without and with synchronization are given. Figures 13(b) and 13(c) show the x - z and y - z planes for the same 4-scroll attractor. Synchronization error is represented as a straight line in phase planes x - \hat{x} , y - \hat{y} , and z - \hat{z} as shown in Figure 13(d).

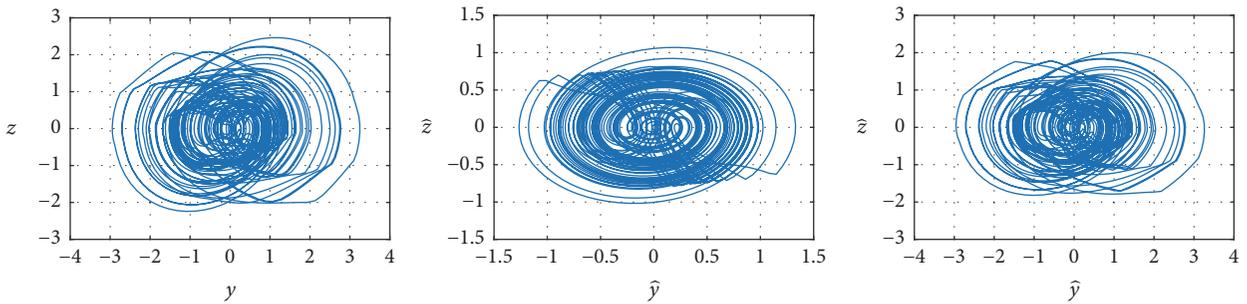
6.2. Synchronization of Nonoptimized 4-Scroll with Case 1 in Table 6: Different Delays. Similar to previous section, by using different delays for slopes in SNLF (irregular) as given in case 1, Table 6, the resulting 4-scroll chaotic attractor can be synchronized with another 4-scroll chaotic attractor that preserves traditional values for system's parameters as well as a regular SNLF. For master system, we use different delays $h = -2, 0.5, 2$, slope = 10 (case 1, Table 6) for irregular SNLF, system's parameters $a = 1.0$, $b = 0.642$, $c = 0.514$, $d_1 = 1.0$,



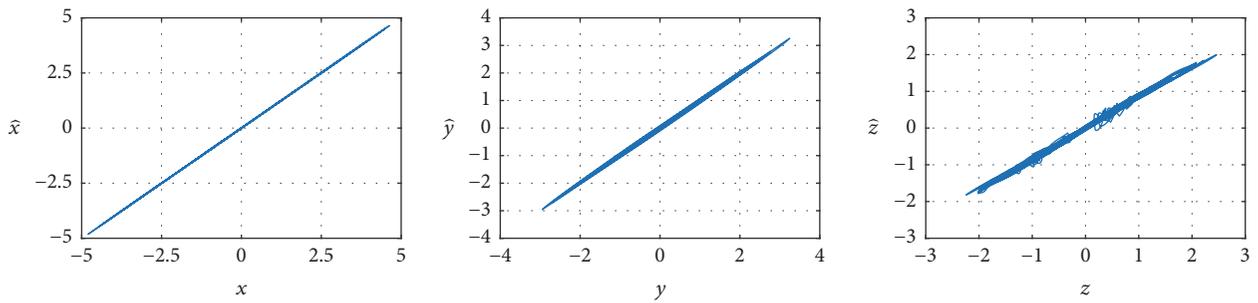
(a) Optimized 4-scroll chaotic system, typical 4-scroll chaotic system without optimization, and typical 4-scroll chaotic system after synchronization (x-y plane)



(b) Optimized 4-scroll chaotic system, typical 4-scroll chaotic system without optimization, and typical 4-scroll chaotic system after synchronization (x-z plane)



(c) Optimized 4-scroll chaotic system, typical 4-scroll chaotic system without optimization, and typical 4-scroll chaotic system after synchronization (y-z plane)



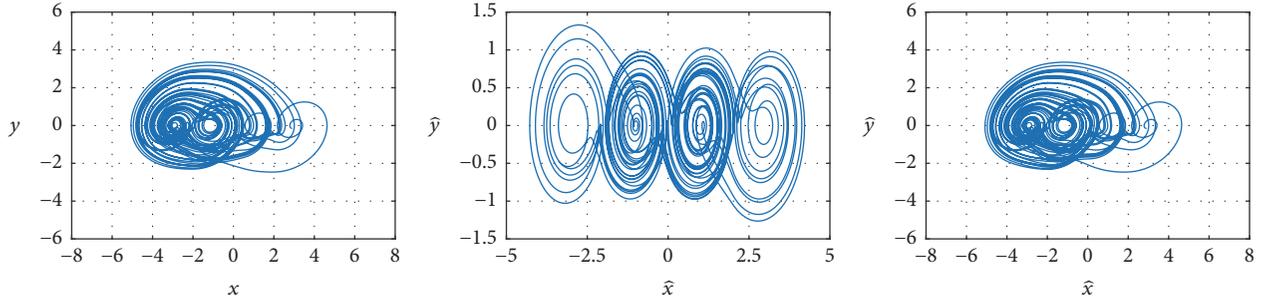
(d) Synchronization error for optimized and nonoptimized 4-scroll chaotic system

FIGURE 13: Synchronization results between optimized and nonoptimized chaotic oscillators with different slopes.

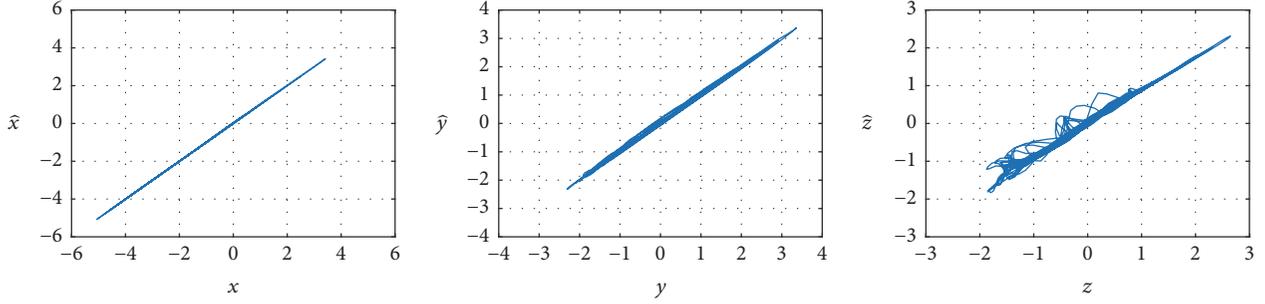
synchronization gains $k_1 = 4$, $k_2 = 20$, $k_3 = 45$, and initial conditions $ic = 2.99, -0.03, 0.1$. On the other hand, for slave system traditional values $a = 0.7$, $b = 0.7$, $c = 0.7$, $d_1 = 0.7$ are considered; also a symmetrical SNLF with slopes = 200 and delays $h = -2, 0, 2$ are used, $ic = 2.9, 0.01, 0.01$. Figure 14 shows the simulation results of synchronization.

In Figure 14(a) the phase portraits for master system and slave system without and with synchronization are given. Synchronization error between $x-\hat{x}$, $y-\hat{y}$, and $z-\hat{z}$ is given in Figure 14(b).

Next, Figure 15 shows the transient response for synchronization. In both cases, the numerical simulations confirm an



(a) Optimized 4-scroll chaotic system, typical 4-scroll chaotic system without optimization, and typical 4-scroll chaotic system after synchronization (x - y plane)



(b) Synchronization error for optimized and nonoptimized 4-scroll chaotic system

FIGURE 14: Synchronization results between optimized and nonoptimized chaotic oscillators with different delays of the slopes.

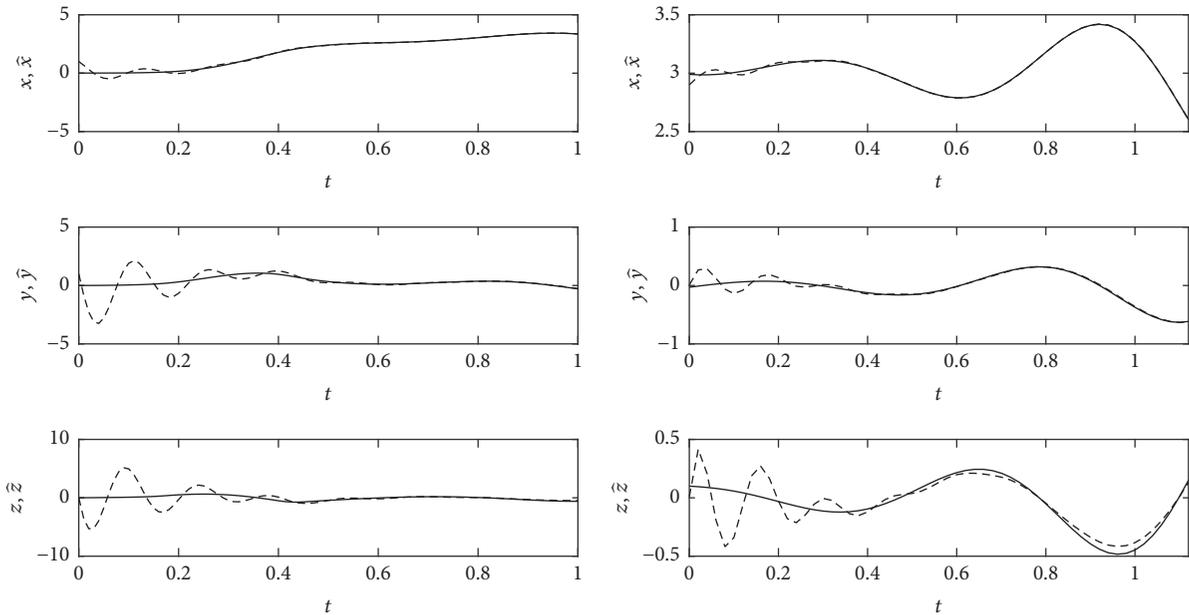


FIGURE 15: Transient response for both cases in Figures 13 and 14, respectively.

adequate synchronization between optimized and nonoptimized multiscroll chaotic oscillators.

As a general result, the experimental realizations of optimized multiscroll chaotic oscillators confirmed that the irregular SNLF provides a suitable mechanism to obtain a positive LE with a higher magnitude than without optimization as shown in Tables 4 and 6. In addition, the

irregular SNLF proposed in this paper leads to getting a high value of the positive LE when compared to other approaches also using evolutionary algorithms to maximize the positive LE as shown in Table 7. The main characteristics, such as maximum value of the positive LE, experimental realization, and synchronization, of this work are given in Table 7.

TABLE 7: Comparison of the main results of this work with other approaches.

Characteristic	Ref [22]	Ref [19]	This work
Multiscrolls	✓ (2 to 9 scrolls)	✓ (2 to 6 scrolls)	✓ (2 to 5 scrolls)
Optimization algorithm	✓ (DE)	✓ (DE)	✓ (DE)
Type of SNLF	Regular (symmetrical)	Regular (symmetrical)	Irregular (asymmetrical)
Maximum value of the positive LE after optimization (5 scrolls)	0.4416	0.2711	0.7112 (case 3 in Table 4)
Experimental verification	–	–	✓ (2 to 5 scrolls)
Synchronization scheme	–	–	✓ (4 scrolls)

7. Conclusions

It has been shown the experimental realization of optimized multiscrolls chaotic attractors. More specifically, 2 to 5 scrolls were obtained by using irregular SNLFs. Two cases were considered: variation of the slopes of SNLF and variation of the delays of slopes. Both cases led us to implement irregular SNLFs and, by applying an optimization evolutionary algorithm, the positive LE was maximized. From the experimental results, it can be appreciated that the higher value of positive Lyapunov exponents is obtained by varying the slopes of SNLF. Also, all the experiments listed in the last two sections confirmed that the value of positive LE increases by augmenting the number of scrolls. On the other hand, the synchronization between optimized and nonoptimized chaotic attractors was demonstrated. Further, this research could be useful to propose engineering applications based on chaos, for example, secure communications, since the obtained results have been performed at experimental circuit level.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been partially supported by CONACYT-SNI (Mexico). The authors thankfully acknowledge the computer resources, technical expertise, and support provided by the “Laboratorio Nacional de Supercómputo del Sureste de México (National Laboratory of High Performance Computing),” belonging to CONACYT Network of National Laboratories. J. M. Muñoz-Pacheco acknowledges CONACYT for the financial support (no. 258880: Proyecto Apoyado por el Fondo Sectorial de Investigación para la Educación).

References

- [1] J. C. Sprott, “New chaotic regimes in the Lorenz and Chen systems,” *International Journal of Bifurcation and Chaos*, vol. 25, no. 2, Article ID 1550033, 2015.
- [2] T. Gotthans and J. Petržela, “New class of chaotic systems with circular equilibrium,” *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1143–1149, 2015.
- [3] Z. Wang, J. Ma, Z. Chen, and Q. Zhang, “A new chaotic system with positive topological entropy,” *Entropy*, vol. 17, no. 8, pp. 5561–5579, 2015.
- [4] V.-T. Pham, C. Volos, S. Jafari, Z. Wei, and X. Wang, “Constructing a novel no-equilibrium chaotic system,” *International Journal of Bifurcation and Chaos*, vol. 24, no. 5, Article ID 1450073, 2014.
- [5] D. Chen, Z. Sun, X. Ma, and L. Chen, “Circuit implementation and model of a new multi-scroll chaotic system,” *International Journal of Circuit Theory and Applications*, vol. 42, no. 4, pp. 407–424, 2014.
- [6] P. Muthukumar, P. Balasubramaniam, and K. Ratnavelu, “Fast projective synchronization of fractional order chaotic and reverse chaotic systems with its application to an affine cipher using date of birth (DOB),” *Nonlinear Dynamics*, vol. 80, no. 4, pp. 1883–1897, 2015.
- [7] Y. Xu, H. Wang, Y. Li, and B. Pei, “Image encryption based on synchronization of fractional chaotic systems,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 10, pp. 3735–3744, 2014.
- [8] X.-Y. Ji, S. Bai, Y. Guo, and H. Guo, “A new security solution to JPEG using hyper-chaotic system and modified zigzag scan coding,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 22, no. 1-3, pp. 321–333, 2015.
- [9] H.-I. Hsiao and J. Lee, “Fingerprint image cryptography based on multiple chaotic systems,” *Signal Processing*, vol. 113, pp. 169–181, 2015.
- [10] L. J. Ontanon-Garcia, E. Jimenez-Lopez, E. Campos-Canton, and M. Basin, “A family of hyperchaotic multi-scroll attractors in R^n ,” *Applied Mathematics and Computation*, vol. 233, pp. 522–533, 2014.
- [11] J. Ma, Z. Chen, Z. Wang, and Q. Zhang, “A four-wing hyperchaotic attractor generated from a 4D memristive system with a line equilibrium,” *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1275–1288, 2015.
- [12] E. Zambrano-Serrano, E. Campos-Cantón, and J. M. Muñoz-Pacheco, “Strange attractors generated by a fractional order switching system and its topological horseshoe,” *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1629–1641, 2016.
- [13] S. Hassan Hosseinnia, R. L. Magin, and B. M. Vinagre, “Chaos in fractional and integer order NSG systems,” *Signal Processing*, vol. 107, pp. 302–311, 2015.
- [14] Y. Xu, R. Gu, H. Zhang, and D. Li, “Chaos in diffusionless Lorenz system with a fractional order and its control,” *International Journal of Bifurcation and Chaos*, vol. 22, no. 4, Article ID 1250088, 2012.

- [15] Y. Xu, H. Wang, D. Liu, and H. Huang, "Sliding mode control of a class of fractional chaotic systems in the presence of parameter perturbations," *Journal of Vibration and Control*, vol. 21, no. 3, pp. 435–448, 2015.
- [16] D. Liu, W. Xu, and Y. Xu, "Noise-induced chaos in the elastic forced oscillators with real-power damping force," *Nonlinear Dynamics*, vol. 71, no. 3, pp. 457–467, 2013.
- [17] Y. Xu, R. Gu, and H. Zhang, "Effects of random noise in a dynamical model of love," *Chaos, Solitons & Fractals*, vol. 44, no. 7, pp. 490–497, 2011.
- [18] Y. Xu, G. M. Mahmoud, W. Xu, and Y. Lei, "Suppressing chaos of a complex Duffing's system using a random phase," *Chaos, Solitons & Fractals*, vol. 23, no. 1, pp. 265–273, 2005.
- [19] L. G. De La Fraga and E. Tlelo-Cuautle, "Optimizing the maximum Lyapunov exponent and phase space portraits in multi-scroll chaotic oscillators," *Nonlinear Dynamics*, vol. 76, no. 2, pp. 1503–1515, 2014.
- [20] F. L. Dubeibe and L. D. Bermúdez-Almanza, "Optimal conditions for the numerical calculation of the largest Lyapunov exponent for systems of ordinary differential equations," *International Journal of Modern Physics C*, vol. 25, no. 7, Article ID 1450024, 2014.
- [21] M. Ciobanu, A. Ardelean, C. Cotoraci, and L. Mos, "Maximum lyapunov exponents evidencing chaos in neural activity," *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 11, pp. 2600–2603, 2013.
- [22] V. H. Carbajal-Gomez, E. Tlelo-Cuautle, and F. V. Fernandez, "Optimizing the positive Lyapunov exponent in multi-scroll chaotic oscillators with differential evolution algorithm," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8163–8168, 2013.
- [23] C. Sanchez-Lopez, "Automatic synthesis of chaotic attractors," *Applied Mathematics and Computation*, vol. 217, no. 9, pp. 4350–4358, 2011.
- [24] G. Ablay, "Novel chaotic delay systems and electronic circuit solutions," *Nonlinear Dynamics*, vol. 81, no. 4, pp. 1795–1804, 2015.
- [25] M. Shahzad, V.-T. Pham, M. A. Ahmad, S. Jafari, and F. Hadaeghi, "Synchronization and circuit design of a chaotic system with coexisting hidden attractors," *The European Physical Journal Special Topics*, vol. 224, no. 8, pp. 1637–1652, 2015.
- [26] J. Ma, X. Y. Wu, R. T. Chu, and L. Zhang, "Selection of multi-scroll attractors in Jerk circuits and their verification using Pspice," *Nonlinear Dynamics*, vol. 76, no. 4, pp. 1951–1962, 2014.
- [27] J. M. Muñoz-Pacheco, E. Tlelo-Cuautle, I. Toxqui-Toxqui, C. Sánchez-López, and R. Trejo-Guerra, "Frequency limitations in generating multi-scroll chaotic attractors using CFOAs," *International Journal of Electronics*, vol. 101, no. 11, pp. 1559–1569, 2014.
- [28] Y. Ma, Y. Li, and X. Jiang, "Simulation and circuit implementation of 12-scroll chaotic system," *Chaos, Solitons & Fractals*, vol. 75, pp. 127–133, 2015.
- [29] J. B. Dingwell, *Lyapunov Exponents. Wiley Encyclopedia of Biomedical Engineering*, John Wiley & Sons, 2006.
- [30] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.
- [31] R. Hegger, H. Kantz, and T. Schreiber, "Practical implementation of nonlinear time series methods: the TISEAN package," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 9, no. 2, pp. 413–435, 1999.
- [32] N. Look, C. J. Arellano, A. M. Grabowski, W. J. McDermott, R. Kram, and E. Bradley, "Dynamic stability of running: The effects of speed and leg amputations on the maximal lyapunov exponent," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 23, no. 4, Article ID 043131, 2013.
- [33] A. Sarkar, A. Narváez, and J. Harting, "Quantification of the performance of chaotic micromixers on the basis of finite time Lyapunov exponents," *Microfluidics and Nanofluidics*, vol. 13, no. 1, pp. 19–27, 2012.
- [34] H. Liu, A. Abraham, and M. Clerc, "Chaotic dynamic characteristics in swarm intelligence," *Applied Soft Computing*, vol. 7, no. 3, pp. 1019–1026, 2007.
- [35] D. Yang, Z. Liu, and J. Zhou, "Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 1229–1246, 2014.
- [36] X. Ai, K. Sun, S. He, and H. Wang, "Design of grid multiscroll chaotic attractors via transformations," *International Journal of Bifurcation and Chaos*, vol. 25, no. 10, Article ID 1530027, 2015.
- [37] V.-T. Pham, C. K. Volos, and S. Vaidyanathan, "Multi-scroll chaotic oscillator based on a first-order delay differential equation," *Studies in Computational Intelligence*, vol. 581, pp. 59–72, 2015.
- [38] J. Lü and G. Chen, "Generating multiscroll chaotic attractors: theories, methods and applications," *International Journal of Bifurcation and Chaos*, vol. 16, no. 4, pp. 775–858, 2006.
- [39] J. M. Muñoz-Pacheco, E. Zambrano-Serrano, O. Félix-Beltrán, L. C. Gómez-Pavón, and A. Luis-Ramos, "Synchronization of PWL function-based 2D and 3D multi-scroll chaotic systems," *Nonlinear Dynamics*, vol. 70, no. 2, pp. 1633–1643, 2012.

Research Article

Improved Hybrid Fireworks Algorithm-Based Parameter Optimization in High-Order Sliding Mode Control of Hypersonic Vehicles

Xiaomeng Yin, Xing Wei, Lei Liu , and Yongji Wang

National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Lei Liu; liulei@hust.edu.cn

Received 29 December 2017; Accepted 31 January 2018; Published 4 March 2018

Academic Editor: László T. Kóczy

Copyright © 2018 Xiaomeng Yin 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.

With respect to the nonlinear hypersonic vehicle (HV) dynamics, achieving a satisfactory tracking control performance under uncertainties is always a challenge. The high-order sliding mode control (HOSMC) method with strong robustness has been applied to HVs. However, there are few methods for determining suitable HOSMC parameters for an efficacious control of HV, given that the uncertainties are randomly distributed. In this study, we introduce a hybrid fireworks algorithm- (FWA-) based parameter optimization into HV control design to satisfy the design requirements with high probability. First, the complex relation between design parameters and the cost function that evaluates the likelihood of system instability and violation of design requirements is modeled via stochastic robustness analysis. Subsequently, we propose an efficient hybrid FWA to solve the complex optimization problem concerning the uncertainties. The efficiency of the proposed hybrid FWA-based optimization method is demonstrated in the search of the optimal HV controller, in which the proposed method exhibits a better performance when compared with other algorithms.

1. Introduction

Hypersonic vehicles (HVs) have attracted increasing interest given their characteristics of high speed and excellent cost effectiveness to access the space. HVs usually fight in near space at a high speed, in which the aerodynamic properties are difficult to predict [1]. Additionally, owing to the peculiar structure of HVs, the couplings related to aerodynamics, propulsion, and structural dynamics are strong, and this makes HV sensitive to uncertainties [2]. In this study, we focus on the efficacious control design of nonlinear HV dynamics given that uncertainties are randomly distributed.

As members of sliding mode control methods [3–5], high-order sliding mode control (HOSMC) methods [6–8] exhibit strong robustness and a reduced chattering effect while dealing with uncertainties. For example, Zhang et al. [8] proposed a quasi-continuous HOSMC for HV to effectively alleviate the chattering phenomena. In addition to the chattering effect, several design requirements also

should be considered for practical HV control under the effects of uncertainties. The priority is guaranteeing the stability. Furthermore, in order to ensure a satisfactory control performance, high-accuracy tracking of trajectory commands and lower fuel consumption are desired. However, when uncertainties are involved in the nonlinear control structure of HV, it is a challenge to adjust design parameters to reach a satisfied level of tracking performance. Two problems have appeared because of introducing uncertainties into the HOSM control of HV.

The first problem is that the modeling of the relation between the design parameters and the HV tracking performance under the effect of uncertain parameters is complex. Dealing with uncertainty in a probabilistic way, stochastic robustness analysis (SRA) was first proposed by Stengel and Ray [9], and it is an effective method to evaluate the extent to which the specified design requirements are satisfied. A cost function for SRA is formulated to estimate the likelihood that the design requirements are not satisfied. Subsequently,

the design parameter space is searched to minimize the cost function to obtain the optimal performance in the presence of uncertainties [10]. Cao et al. [11] optimized the HV controller parameters by using SRA and hybrid PSO algorithm. However, only the dynamic response indices of step command were concerned in the cost function for SRA [11–13]. In order to achieve a desired tracking performance despite uncertainties, it is necessary to introduce appropriate indices that characterize the command tracking process and corresponding indicator functions into the optimization problem modeling of HV.

The second important problem in the HOSM control of HV involves solving the optimization problem. Conventional optimization methods, such as the gradient search method, are no longer suitable given that the partial derivative of the cost function in SRA is difficult to obtain. For complex optimization problem involving uncertainties, a high efficiency computational intelligence optimization algorithm is required to determine the optimal controller parameters of HV to achieve a satisfied level of tracking performance under the influence of uncertainties. Nowadays, various computational intelligence techniques [14, 15], such as genetic algorithm (GA) [16], particle swarm optimization (PSO) [17], and differential evolutionary (DE), have been proposed for complex optimization problems with the development of computation technology.

Among computational algorithms, the fireworks algorithm (FWA) is a relatively new swarm intelligence-based algorithm proposed by Tan and Zhu [18]. It simulates the process of fireworks explosion, in which the “good” fireworks generate more sparks in smaller explosion areas. Numerical experiments indicated that FWA converges to a global optimum with a smaller number of function evaluations than PSO and GA [19]. Li et al. [20] proposed an adaptive fireworks algorithm (AFWA) in which the explosion amplitude of fireworks that fails to produce a better spark increases. To improve interaction of solutions, hybrid algorithm of FWA-DE was developed by Zheng et al. [21]. Zhang et al. [22] proposed an improved FWA by enhancing fireworks interaction. With respect to improvements in the FWA [20–23], it is recognized that the diversification mechanism of FWA does not utilize more information on other qualified solutions in the swarm. Therefore, with respect to the HV control under uncertainties that are randomly distributed, it is necessary to develop an improved FWA with enhanced solutions interaction to effectively solve the complex optimization problem of searching for the optimal controller.

In this study, an improved hybrid FWA-based parameter optimization method is proposed for HV control to achieve an excellent tracking performance in the presence of uncertainties. The main contributions are as follows:

(1) The uncertainties that are randomly distributed are considered in the modeling phase via SRA. The cost function evaluating the probability of design requirements violation is formulated to model the complex relation between design parameters and tracking performance of the uncertain HV system. Appropriate indices of the command tracking response are developed.

(2) A hybrid FWA to search for the optimal design parameters is proposed for the complex optimization problem involving uncertainties to satisfy design requirements with high probability. The introduction of the hybrid FWA into SRA effectively optimizes the tracking performance of the nonlinear HV system under uncertainties.

This study is organized as follows: In Section 2, the optimization problem in the HOSM control of HV is introduced. In Section 3, the complex relation between design parameters and HV performance under uncertainties is modeled. Section 4 proposes a new hybrid FWA to determine the optimal parameters of HV. Section 5 investigates the global convergence of the proposed hybrid FWA, and the simulation and comparison results are demonstrated. A few conclusions are made in Section 6.

2. HOSM Control Structure of HV with Uncertainties

The control-oriented model of a generic hypersonic vehicle (HV) is described by [24]. An inverse-square-law gravitational model and centripetal acceleration are considered, and the dynamic differential equations for velocity V , altitude h , flight-path angle γ , angle of attack α , and pitch rate q of HV are as follows:

$$\begin{aligned}\dot{V} &= \frac{T \cos \alpha - D}{m} - \frac{\mu \sin \gamma}{r^2}, \\ \dot{h} &= V \sin \gamma, \\ \dot{\gamma} &= \frac{L + T \sin \alpha}{mV} - \frac{(\mu - V^2 r) \cos \gamma}{Vr^2}, \\ \dot{\alpha} &= q - \dot{\gamma}, \\ \dot{q} &= \frac{M_{yy}}{I_{yy}},\end{aligned}\quad (1)$$

with

$$\begin{aligned}L &= \frac{1}{2} \rho V^2 S C_L(\alpha), \\ D &= \frac{1}{2} \rho V^2 S C_D(\alpha), \\ T &= \frac{1}{2} \rho V^2 S C_T(\beta), \\ M_{yy} &= \frac{1}{2} \rho V^2 S \bar{c} [C_M(\alpha) + C_M(\alpha, \delta_e) + C_M(\alpha, q)], \\ r &= h + R_E,\end{aligned}\quad (2)$$

where L is the lift, D is the drag, T is the thrust, and M_{yy} is the pitching moment. m , r , R_E , μ , and ρ denote the mass, radial distance, radius of the Earth, gravitational constant, and density of air, respectively. Additionally, S , \bar{c} , and I_{yy} denote the reference area, mean aerodynamic chord, and the moment of inertia about y -body axes, respectively. δ_e denotes the elevator deflection, and β denotes the engine throttle setting.

The thrust T in (2) is provided by the engine dynamics, and this is represented as follows [10]:

$$\ddot{\beta} = k_1 \dot{\beta} + k_2 \beta + k_3 \beta_c, \quad (3)$$

where β_c denotes the engine throttle setting command. It is adopted that $k_1 = k_2 = 0$ and $k_3 = 1$ for proper modeling of engine dynamics.

In order to guarantee the robustness of the HV flight control system, the parametric uncertainties in (1)-(2) are considered as follows:

$$\begin{aligned} m &= m_0 (1 + \Delta m), \\ \rho &= \rho_0 (1 + \Delta \rho), \\ C_L &= C_{L0} (1 + \Delta C_L), \\ C_D &= C_{D0} (1 + \Delta C_D), \\ C_T &= C_{T0} (1 + \Delta C_T), \\ C_M &= C_{M0} (1 + \Delta C_M), \end{aligned} \quad (4)$$

where the uncertainties Δm , $\Delta \rho$, ΔC_L , ΔC_D , ΔC_T , and ΔC_M are bounded.

HV system (1) with engine dynamics is highly nonlinear. The relationship between input variables $u = [\delta_e, \beta_c]^T$ and the output variables $y = [V, h]^T$ is apparently expressed by the feedback linearization method [10]. We differentiate V three times and differentiate h four times, and we obtain the following expressions:

$$\begin{aligned} \dot{V} &= \frac{(T \cos \alpha - D)}{m} - \frac{\mu \sin \gamma}{r^2} \triangleq f(x), \\ \ddot{V} &= \frac{\Omega_1 \dot{x}}{m}, \end{aligned} \quad (5)$$

$$\ddot{\ddot{V}} = \frac{(\Omega_1 \ddot{x} + \dot{x}^T \Omega_2 \dot{x})}{m},$$

$$\dot{h} = V \sin \gamma,$$

$$\ddot{h} = \dot{V} \sin \gamma + V \dot{\gamma} \cos \gamma, \quad (6)$$

$$\ddot{\ddot{h}} = \ddot{V} \sin \gamma + 2\dot{V} \dot{\gamma} \cos \gamma - V \dot{\gamma}^2 \sin \gamma + V \ddot{\gamma} \cos \gamma,$$

$$\begin{aligned} h^{(4)} &= \ddot{\ddot{V}} \sin \gamma + 3\dot{V} \ddot{\gamma} \cos \gamma - 3\dot{V} \dot{\gamma}^2 \sin \gamma + 3\dot{V} \ddot{\gamma} \cos \gamma \\ &\quad - 3V \dot{\gamma} \ddot{\gamma} \sin \gamma - V \dot{\gamma}^3 \cos \gamma + V \ddot{\ddot{\gamma}} \cos \gamma, \end{aligned} \quad (7)$$

$$\begin{aligned} \dot{\gamma} &= \frac{(L + T \sin \alpha) / m}{V} - \frac{(\mu - V^2 r) \cos \gamma / V}{r^2} \triangleq g(x), \\ \ddot{\gamma} &= \pi_1 \dot{x}, \end{aligned} \quad (8)$$

$$\ddot{\ddot{\gamma}} = \pi_1 \ddot{x} + \dot{x}^T \pi_2 \dot{x},$$

where $x = [V, \gamma, \alpha, \beta, h]^T$, $\Omega_1 = \partial f(x) / \partial x$, $\Omega_2 = \partial \Omega_1 / \partial x$, and $\pi_1 = \partial g(x) / \partial x$, $\pi_2 = \partial \pi_1 / \partial x$.

In order to force the velocity and altitude to track the time-varying commanded output $y_c = [V_c, h_c]^T$, we define

the velocity sliding tracking error and the altitude sliding tracking error as $\sigma_v = V - V_c$ and $\sigma_h = h - h_c$, respectively. Based on (5) and (7), we have

$$\begin{bmatrix} \ddot{\sigma}_v \\ \sigma_h^{(4)} \end{bmatrix} = F(x) + G(x) u, \quad (9)$$

where the formulations of F , G , Ω_1 , Ω_2 , π_1 , and π_2 are the same as those in [10].

As stated in [10], the matrix $G(x)$ in (9) is nonsingular over the entire flight envelope of HV, so (9) is decoupled with the auxiliary control input v as follows:

$$\begin{aligned} u &= G(x)^{-1} (-F(x) + v), \\ v &= [v_1 \ v_2]^T = [\ddot{\sigma}_v \ \sigma_h^{(4)}]^T. \end{aligned} \quad (10)$$

A previous study [6] indicates that if appropriate control parameters are designed, then the finite time stabilization of system (9) is guaranteed by the quasi-continuous HOSMC v_1 and v_2 , and this is given as follows:

$$\begin{aligned} v_1 &= \frac{-\alpha_{\xi v} \varphi_{2,3}}{N_{2,3}}, \\ v_2 &= \frac{-\alpha_{\xi h} \varphi_{3,4}}{N_{3,4}}, \end{aligned} \quad (11)$$

with

$$\begin{aligned} \varphi_{2,3} &= \ddot{\sigma}_v + \beta_{\xi v 2} \left(|\dot{\sigma}_v| + \beta_{\xi v 1} |\sigma_v|^{2/3} \right)^{-1/2} \left(\dot{\sigma}_v \right. \\ &\quad \left. + \beta_{\xi v 1} |\sigma_v|^{2/3} \text{sign}(\sigma_v) \right), \\ N_{2,3} &= |\ddot{\sigma}_v| + \beta_{\xi v 2} \left(|\dot{\sigma}_v| + \beta_{\xi v 1} |\sigma_v|^{2/3} \right)^{1/2}, \\ \varphi_{3,4} &= \ddot{\sigma}_h + \beta_{\xi h 3} \left(|\ddot{\sigma}_h| \right. \\ &\quad \left. + \beta_{\xi h 2} \left(|\dot{\sigma}_h| + \beta_{\xi h 1} |\sigma_h|^{3/4} \right)^{2/3} \right)^{-1/2} \left(\ddot{\sigma}_h \right. \\ &\quad \left. + \beta_{\xi h 2} \left(|\dot{\sigma}_h| + \beta_{\xi h 1} |\sigma_h|^{3/4} \right)^{-1/3} \right. \\ &\quad \left. \cdot \left(\dot{\sigma}_h + \beta_{\xi h 1} |\sigma_h|^{3/4} \text{sign}(\sigma_h) \right) \right), \\ N_{3,4} &= |\ddot{\sigma}_h| + \beta_{\xi h 3} \left(|\ddot{\sigma}_h| \right. \\ &\quad \left. + \beta_{\xi h 2} \left(|\dot{\sigma}_h| + \beta_{\xi h 1} |\sigma_h|^{3/4} \right)^{2/3} \right)^{1/2}. \end{aligned} \quad (12)$$

The HV control structure based on HOSM is shown in Figure 1.

For the quasi-continuous HOSM controller (11), the design parameters $\alpha_{\xi i}$ and $\beta_{\xi ij}$ ($i = h, v$, $j = 1, 2, \dots$) define the output trajectory of the HV system, which is shown in Figure 2. In the figure, altitude commands are in the dotted

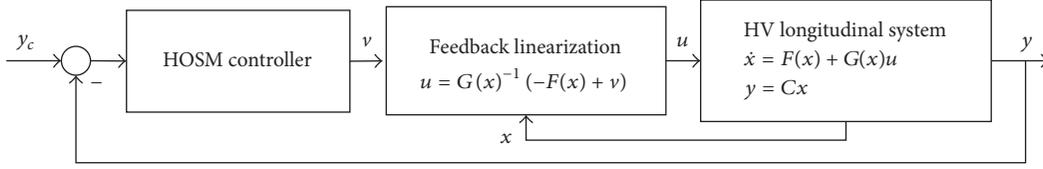


FIGURE 1: HV control structure based on HOSM.

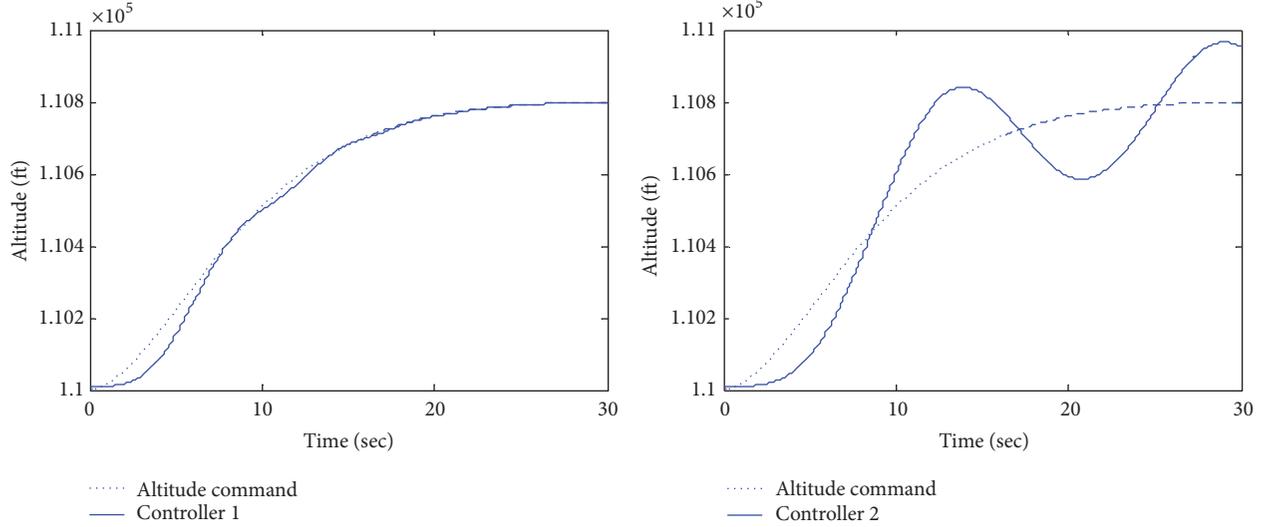


FIGURE 2: Altitude tracking trajectories using different design parameters.

lines, and tracking trajectories are in the solid lines. In order to satisfy the design requirements, it is necessary to optimize the controller parameters.

Furthermore, it is more appealing to satisfy the HV control design requirements under the effects of uncertainties. In Figure 3, the tracking trajectories with uncertain parameters generated randomly are depicted by the solid lines, and the altitude commands are shown by the dotted lines. Within two dashed lines are the trajectories that meet the design requirements.

The simulations indicate that the same set of design parameters will generate various trajectories in the presence of uncertainties. Therefore, it is necessary to employ a proper measure to quantify a set of data values. In this study, the probability that the design requirements are not satisfied is used for the HV performance evaluation with uncertainties.

Therefore, the target of HV control design involves determining the optimal HOSM control parameters to satisfy design requirements with high probability. It is necessary to solve the following two problems in the HOSM parameter optimization: (1) to develop a cost function that evaluates the likelihood of system instability and the violation of the design requirements, so that the complex relation between HV design parameters and the performance under uncertainties is modeled; (2) to solve the complex optimization problem related to the uncertainties by a high efficient computational intelligence optimization algorithm.

3. Stochastic Robustness Analysis of HV

The concept of stochastic robustness was proposed by Stengel and Ray [9], and this is effective in evaluating the extent to which the specified design requirements are satisfied. We deal with uncertainties in a probabilistic way, and thus a cost function to evaluate the likelihood of system instability and the violation of design requirements is formulated via SRA.

The flowchart of HOSM control design for HV based on SRA is shown in Figure 4.

In Figure 4, a closed-loop HV system with uncertain parameters \tilde{u}_k is denoted by the dotted box. $I[\cdot]$ denotes the indicator function corresponding to the design requirement. The value of $I[\cdot]$ is within $[0, 1]$, and this is 0 if an acceptable performance appears and is 1 otherwise.

With the indicator function $I[\cdot]$, the probability of satisfying a certain performance requirement P is defined by an integral of the corresponding indicator function over the expected variation space of parametric uncertainties. It is a practical method to estimate the probability P by Monte Carlo evaluation (MCE) as follows:

$$\hat{P}(d) = \frac{1}{N} \sum_{k=1}^N I[H(\tilde{u}_k), C(d)], \quad \tilde{u}_k \in Q, \quad (13)$$

where $H(\tilde{u}_k)$ represents the HV system with uncertain parameters \tilde{u}_k that are randomly selected within the parameter space Q . $C(d)$ represents the HOSM controller with

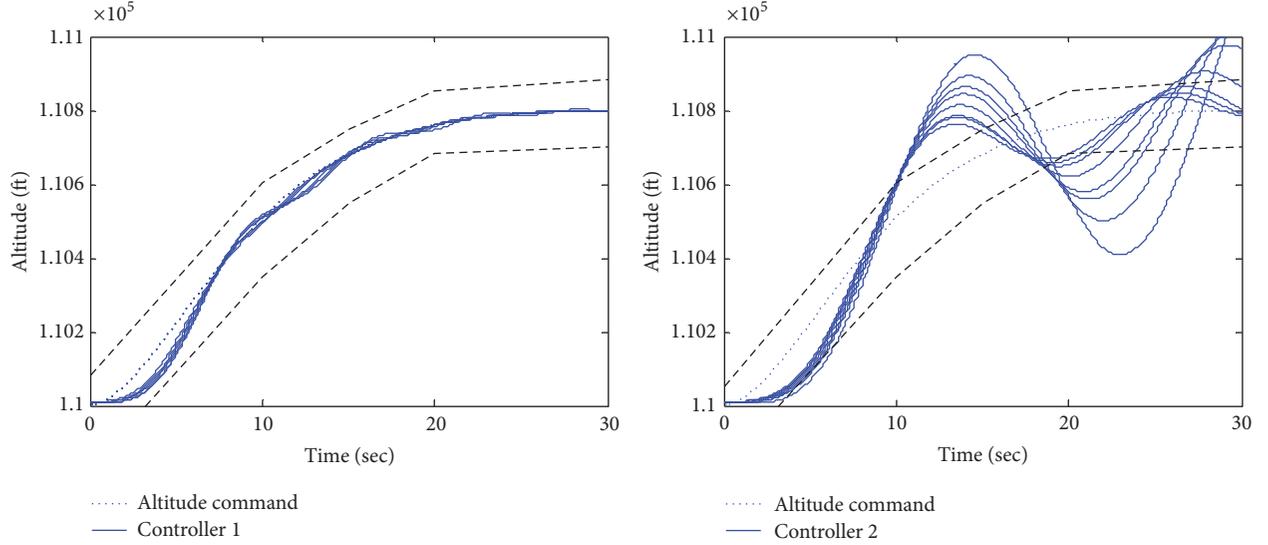


FIGURE 3: Altitude tracking trajectories under uncertainties.

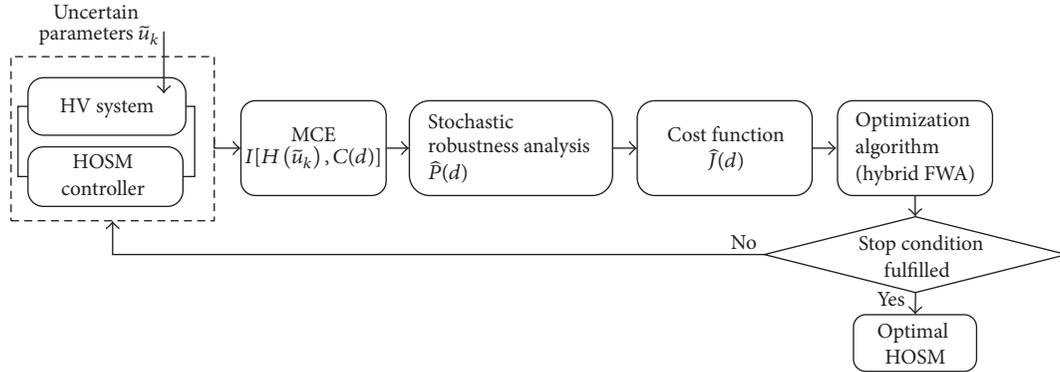


FIGURE 4: Flowchart of HOSM control design based on SRA.

the design parameter vector d , and N denotes the sampling numbers.

Thus, the cost function $\hat{J}(d)$ for SRA is formed by combining the probability of various design requirements with weights as follows:

$$\hat{J}(d) = f(\hat{P}_1(d), \hat{P}_2(d), \dots), \quad (14)$$

where the estimated value of the cost function $\hat{J}(d)$ approaches the true value J when the sampling number $N \rightarrow \infty$.

As shown in Figure 4, the optimal design parameters of HV are determined under the guidance of the cost function $\hat{J}(d)$. Therefore, it is vital to define appropriate stochastic robustness measurements for the cost function to achieve the desired tracking performance despite uncertainties.

3.1. Stochastic Robustness Indices and Indicators. In this section, the stochastic robustness indices and indicators are introduced to evaluate the HV tracking performance in the presence of uncertainties.

According to the requirements of HV control design, the first index is set to guarantee system stability in the presence of uncertainties. Additionally, it is necessary to develop performance indices to characterize the command tracking trajectories of HV. The tracking trajectory of a general reference signal is not standardized as that of step signal, and thus common indices, such as setting time, overshoot, and steady error, are no longer suitable. Thus, the following performance indices are introduced.

(i) Transient tracking performance:

$$\begin{aligned} \text{TTP}_h &= \frac{1}{t_f} \int_0^{t_f} e^{-a_h t} |h(t) - h_c(t)| \cdot dt, \\ \text{TTP}_v &= \frac{1}{t_f} \int_0^{t_f} e^{-a_v t} |V(t) - V_c(t)| \cdot dt, \end{aligned} \quad (15)$$

where TTP_h and TTP_v represent the transient tracking performance indices for the altitude response and the velocity response, respectively. t_f is the terminal time of the tracking command. a_h and a_v are small

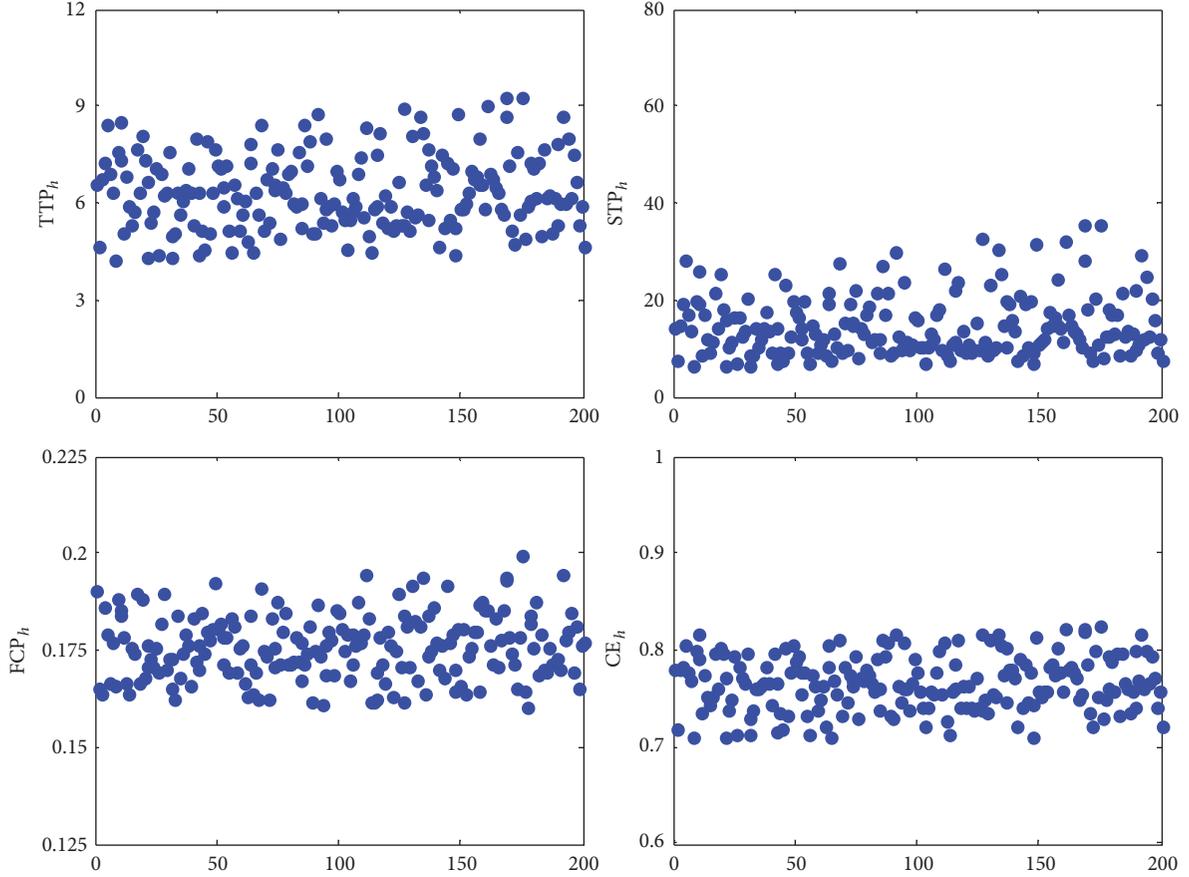


FIGURE 5: Distribution of index values.

positive constants that define the duration of the interested transient stage. A decrease in the value of TTP decreases the tracking error in the transient stage.

(ii) Steady tracking performance:

$$\begin{aligned} STP_h &= \frac{1}{t_f} \int_0^{t_f} (1 - e^{-a_h t}) |h(t) - h_c(t)| \cdot dt, \\ STP_v &= \frac{1}{t_f} \int_0^{t_f} (1 - e^{-a_v t}) |V(t) - V_c(t)| \cdot dt, \end{aligned} \quad (16)$$

where STP_h and STP_v represent the steady tracking performance indices for the altitude response and the velocity response, respectively. A decrease in the value of STP decreases the tracking error in the steady stage.

(ii) Fuel consumption performance:

$$FCP_i = \frac{1}{t_f} \int_0^{t_f} |\beta(t)| \cdot dt, \quad i = h, v, \quad (17)$$

where FCP_h and FCP_v represent the fuel consumption performance indices for the altitude response and the velocity response, respectively. $\beta(t)$ denotes engine throttle setting during the flight. It is necessary to limit FCP within reasonable bounds.

(iv) Chattering effect:

$$CE_i = \max(|\delta_e(t)|), \quad t \in [t_s, t_f], \quad i = h, v, \quad (18)$$

where t_s denotes the time when the sliding tracking errors σ_v and σ_h both tend to zero. CE_h and CE_v represent the maximum chatter amplitude of elevator for the altitude and velocity commands when $t \in [t_s, t_f]$, respectively. The chattering effect can severely deteriorate the flight control performance, and thus it is necessary to attenuate it.

Through Monte Carlo sampling, the distribution of aforementioned index values is obtained from the tracking trajectories under uncertainties. In Figure 5, after 200 times of random sampling, the distributions corresponding to the altitude tracking performance indices are shown. In order to evaluate the extent to which the design requirements are satisfied in the presence of uncertainties, the indicator function corresponding to the index should be carefully defined.

The commonly used indicator is a binary function with two values of 0 and 1 to represent whether the design requirement is satisfied or not. However, for a practical engineering system, there exists an interval between the satisfied and unsatisfied performance. Thus, the following

TABLE 1: Stochastic robustness stability and performance indices of HV.

Metric number	Weight in $J(d)$	Indicator function $I[\cdot]$	Design requirements
1 (2)	10.0 (10.0)	I_1 (I_2)	System stability in altitude response (velocity response)
3 (4)	1.0 (1.0)	I_3 (I_4)	Transient tracking performance in altitude (velocity) response is less than $TTP_{h,1}$ ($TTP_{v,1}$)
5 (6)	1.0 (1.0)	I_5 (I_6)	Steady tracking performance in altitude (velocity) response is less than $STP_{h,1}$ ($STP_{v,1}$)
7 (8)	1.0 (1.0)	I_7 (I_8)	Fuel consumption performance in altitude (velocity) response is less than $FCP_{h,1}$ ($FCP_{v,1}$)
9 (10)	0.5 (0.5)	I_9 (I_{10})	Chattering effect in altitude (velocity) response is less than $CE_{h,1}$ ($CE_{v,1}$)

continuous function is employed as the indicator $I[\cdot]$ as follows:

$$I(x) = \begin{cases} 1 & x > x_1 \\ \left(\frac{x - x_2}{x_1 - x_2}\right)^2 & x_2 < x \leq x_1 \\ 0 & x \leq x_2, \end{cases} \quad (19)$$

where x denotes the value of the performance index, such as TTP_i , STP_i , FCP_i , and CE_i , $i = h, v$. The positive constant x_1 represents $TTP_{i,1}$, $STP_{i,1}$, $FCP_{i,1}$, or $CE_{i,1}$. The positive constant x_2 represents $TTP_{i,2}$, $STP_{i,2}$, $FCP_{i,2}$, or $CE_{i,2}$, and x_1 and x_2 are set by the designer to define the interval between the satisfied and unsatisfied performance.

3.2. Optimization Problem. In order to evaluate the HV tracking performance under uncertainties, the aforementioned indices and indicator functions are employed to formulate the cost function $\hat{J}(d)$ in (14), and they are listed in Table 1.

By formulating the cost function $\hat{J}(d)$, the complex relation between the HOSM controller parameters and the HV tracking performance under uncertainties is modeled. The optimal controller parameters are obtained by solving the following optimization problem:

$$\begin{aligned} \text{Minimize} \quad & J(d) = \sum_{i=1}^{10} \omega_i \hat{P}_i(d) \\ \text{subject to} \quad & d = [\alpha_{\xi v}, \beta_{\xi v1}, \beta_{\xi v2}, \alpha_{\xi h}, \beta_{\xi h1}, \beta_{\xi h2}, \beta_{\xi h3}] \\ & \in [d_{\min}, d_{\max}], \end{aligned} \quad (20)$$

where d is the design parameter vector in the HV controller (11), and this is searched within $[d_{\min}, d_{\max}]$. ω_i denotes the weights for the probabilities $\hat{P}_i(d)$ of various design requirements. The weight in cost function allows a trade-off between design requirements.

Optimization problem (20) is a constrained nonlinear and nonconvex optimization problem, in which the cost function value is calculated with the Monte Carlo method. It is very difficult and time-consuming to determine the optimal solution.

Therefore, for complex optimization problem (20) related to the uncertainties, it is necessary to develop a high efficient

computational intelligence optimization algorithm to determine the optimal HV control parameters, so that an excellent tracking performance can be achieved despite uncertainties.

4. Optimization Technique with Improved Hybrid Fireworks Algorithm

In this section, we propose a hybrid FWA to solve the complex optimization problem of determining the optimal HV control parameters under uncertainties. First, by introducing the GA operators into the mutation process of AFWA, a hybrid FWA is developed with an improved diversification mechanism. Subsequently, the process of the hybrid FWA-based parameter optimization method is illustrated.

4.1. Adaptive Fireworks Algorithm. Inspired by the fireworks explosion, FWA is a relatively new swarm intelligence-based algorithm proposed by Tan and Zhu [18]. In FWA, the fireworks and sparks are considered as the potential solutions in the search space, and the explosion is viewed as a local search around the location of fireworks. The FWA converges to a global optimum with a lower number of function evaluations than those of the PSO and GA [19]. Subsequently, the AFWA [20] was developed to improve the local search capability of the best firework.

The search process of AFWA is as follows:

- (1) Initialization: randomly set the initial locations of fireworks.
- (2) Explosion: each firework generates a set of sparks by executing the regular explosion operation.
- (3) Gaussian mutation: select a few fireworks randomly, and execute the Gaussian explosion (mutation) operation on the selected fireworks to generate several sparks.
- (4) Adaptive amplitude calculation: select the best individual as a firework in the next generation, and calculate its adaptive explosion amplitude.
- (5) Selection: randomly select other fireworks from all individuals.
- (6) Return to Step (2) until the stop criterion is fulfilled.

In order to execute the regular explosion operation in Step (2), the number of sparks N_i of each firework is calculated as follows:

$$N_i = \text{round} \left(N_c \cdot \frac{f_{\max} - f(x_i) + \varepsilon}{\sum_{i=1}^n (f_{\max} - f(x_i)) + \varepsilon} \right). \quad (21)$$

The explosion amplitude A_i is as follows:

$$A_i = A_c \cdot \frac{f(x_i) - f_{\min} + \varepsilon}{\sum_{i=1}^n (f(x_i) - f_{\min}) + \varepsilon}, \quad (22)$$

where n is the number of fireworks. N_c and A_c are two parameters that control the number of sparks and explosion amplitude, respectively. $f(x_i)$ represents the fitness value of x_i , and f_{\max} and f_{\min} denote the maximum and minimum values of the cost function among the n fireworks, respectively. A small constant ε is to avoid zero-division error.

In order to avoid the overwhelming effect of the best firework, the bound of the spark number is set as follows:

$$N_i = \begin{cases} N_{\min} & \text{if } N_i < N_{\min} \\ N_{\max} & \text{else if } N_i > N_{\max} \\ N_i & \text{else,} \end{cases} \quad (23)$$

where N_{\max} and N_{\min} are the upper and lower bounds for N_i .

For a D -dimension problem, after the calculation of spark number and explosion amplitude, the location of each spark x_j is obtained by randomly setting approximately half of the dimensions (z dimensions), and for each dimension k , the value x_j^k ($1 \leq k \leq z$, $1 \leq j \leq N_i$) is set based on x_i^k ($1 \leq i \leq n$). Therefore, the locations of the explosion sparks are set as follows:

$$x_j^k = x_i^k + A_i \cdot \text{rand}(-1, 1). \quad (24)$$

In order to maintain the diversity, for a few randomly selected fireworks, approximately half of the dimensions are selected to change. The mutation sparks are generated by adding a Gaussian distribution coefficient to x_i^k as follows:

$$x_j^k = x_i^k + (x^{*k} - x_i^k) \cdot \text{Gaussian}(0, 1), \quad (25)$$

where x^{*k} is the position of k th dimension of the best firework x^* .

If the new locations of the newly generated sparks are beyond the search space, they are mapped within the search space as follows:

$$x_j^k = x^{k,\min} + \text{rand}(0, 1) \cdot (x^{k,\max} - x^{k,\min}), \quad (26)$$

where $x^{k,\max}$ and $x^{k,\min}$ denote the upper and lower bounds of the k th dimension of the search space, respectively.

In order to improve the local search capability, the best individual is selected as a firework in the next generation. It has adaptive explosion amplitude calculated by selecting an individual that satisfies the following conditions: (1) Its fitness is worse when compared with that of the best firework in

the current generation. (2) Its distance to the best individual is minimal among all individuals that satisfy (1). This is expressed as follows:

$$\hat{s} = \arg \min_{s_i} (d(s_i, s^*)), \quad f(s_i) > f(x^*), \quad (27)$$

where s_i denotes all sparks; s^* denotes the best individual among sparks and fireworks. x^* is the best firework in the current generation, and $d(\cdot)$ represents the distance.

The adaptive amplitude of best firework in next generation is calculated as follows:

$$A^*(g+1) = 0.5 \cdot (A^*(g) + \lambda \cdot \|\hat{s} - x^*\|_{\infty}), \quad (28)$$

where $A^*(g)$ and $A^*(g+1)$ are the adaptive amplitude in current generation g and the next generation $g+1$, respectively. λ is a positive constant (usually higher than 1), and $\|\cdot\|_{\infty}$ represents the infinity norm.

The search process indicates that the diversification mechanism of FWA does not utilize more information on all the qualified solutions, and thus it is necessary to enhance the interaction between fireworks and sparks. It is well known that GA is an efficient evolutionary algorithm that performs searches by combining possible solutions in different directions [16]. Additionally, GA exhibits potential parallelism, and thus individuals can be compared simultaneously. Therefore, we introduce GA into the mutation process of AFWA to generate more diverse and fitter solutions.

4.2. Hybrid Fireworks Algorithm with the Genetic Operator.

In order to improve the search efficiency, the main idea in the proposed hybrid FWA involves utilizing all individuals (fireworks and sparks) to generate new individuals. In order to generate more diverse and fitter solutions, another idea involves selecting the father and mother from individuals with different features that correspond to “core individuals” and “noncore individuals.” Core individuals include the best firework and the sparks generated by the best firework. They exhibit better fitness values and closer locations. Noncore individuals include the other “bad” fireworks and sparks generated by them. They are more diverse.

The process of the genetic operator is given as follows:

- (1) Encoding: encode solutions to become chromosomes (individuals) with discrete units termed as genes.
- (2) Recombination pool construction: construct recombination pool with qualified individuals.
- (3) Parent selection: select parents from core individuals and noncore individuals, respectively.
- (4) Crossover and mutation also exist.

The process of the genetic operator is illustrated in Figure 6.

The steps in the genetic operator are stated in detail as follows.

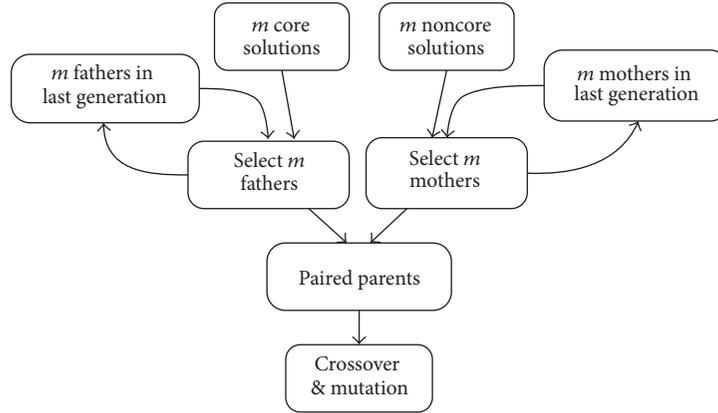


FIGURE 6: Process of the genetic operator.

(i) *Encoding.* The D -dimension solutions are encoded to D -dimension chromosomes, in which each gene represents the value of corresponding dimension of a solution.

(ii) *Recombination Pool Construction.* In order to improve the efficiency of crossover and mutation operations, two pools to select father and mother are constructed. The pool for the selection of the father is constructed by the core individuals from two sources, which include all the fathers (m fathers) in the last generation and several core individuals selected in the current generation (m core individuals). It aids in utilizing the information of the fitter individuals in a wider range. Similarly, the pool for the selection of the mother is constructed by all the mothers (m mothers) in the last generation and several noncore individuals selected in the current generation (m noncore individuals).

With respect to the core individuals that are fitter and located closer, a random selection is applied among them to construct the pool for father selection. Conversely, the noncore individuals are diverse. Therefore, a roulette wheel is employed to select the fitter ones to construct the pool for the selection of the mother.

The algorithm of constructing the recombination pool is shown in Algorithm 1.

(iii) *Parent Selection.* The individuals from the current generation are preferred to select the parents with a higher probability of generating diverse and fitter offspring. In order to select fathers from pool, the fathers of the last generation are replaced by other individuals that have better fitness, and the remaining fathers of the last generation may be replaced by other individuals again with a probability of P_r ($0 < P_r < 1$). Mothers are selected in the same way as the fathers.

The algorithm of selecting parents is shown in Algorithm 2.

(iv) *Crossover and Mutation.* The selected parents are randomly paired to exchange information to generate new two individuals. In the crossover, the tails of a pair of chromosomes (individuals) are swapped at a random point along the gene sequence with a crossover probability P_c ($0.7 <$

$P_c < 1$). After the crossover, the gene in sequence is mutated. This means the offspring are obtained by randomly setting approximately P_m of the dimensions of the individual within the search space, where P_m denotes mutation probability ($0 < P_m < 0.2$).

Thus, a new hybrid FWA is proposed by introducing the GA into the mutation process of AFWA. The flowchart of the proposed optimization algorithm is shown in Figure 7.

Here x^{\max} and x^{\min} are the upper and the lower bounds of the search space, respectively.

4.3. *Hybrid FWA-Based Parameter Optimization.* The proposed hybrid FWA-based parameter optimization method combines the advantages of SRA and the hybrid FWA. By the SRA, the cost function is given to evaluate the HV tracking performance under uncertainties. Subsequently, the hybrid FWA is used to determine the optimal design parameters to satisfy the tracking performance requirements of HV with high probability. The flowchart of the proposed hybrid FWA-based parameter optimization method is given in Figure 8.

To illustrate the search process of the proposed hybrid FWA-based parameter optimization method in detail, the following steps are given:

- (1) Generate several solutions by the hybrid FWA search process.
 - (a) Randomly initialize a population of n fireworks in the search space.
 - (b) For each firework, generate N_i explosion sparks within the explosion amplitude A_i , and subsequently the positions x_j of the explosion sparks are obtained.
 - (c) Encode all individuals as chromosomes.
 - (d) Select $2m$ parents from all the chromosomes, and $2m$ diverse individuals are generated via the genetic operator.
- (2) Evaluate the solution's fitness with the cost function in SRA.

- (1) Construct the pool for the father selection as follows:
 - (1.1) All the fathers of the last generation (m fathers) are reserved in the pool.
 - (1.2) Randomly select m core individuals in the current generation to join the pool.
- (2) Construct the pool for the mother selection as follows:
 - (2.1) All the mothers of the last generation (m mothers) are reserved in the pool.
 - (2.2) Select m noncore individuals in the current generation by the roulette wheel to join the pool.

ALGORITHM 1: Construct the recombination pool.

- (1) Select m fathers from the pool as follows:
 - (1.1) Replace the last generation's fathers by other individuals in pool that have better fitness.
 - (1.2) Replace the remaining last generation's fathers again with a probability P_r ($0 < P_r < 1$).
- (2) Select m mothers from pool as follows:
 - (2.1) Replace the last generation's mothers by other individuals in pool that have better fitness.
 - (2.2) Replace the remaining last generation's mothers again with a probability P_r .

ALGORITHM 2: Select parents.

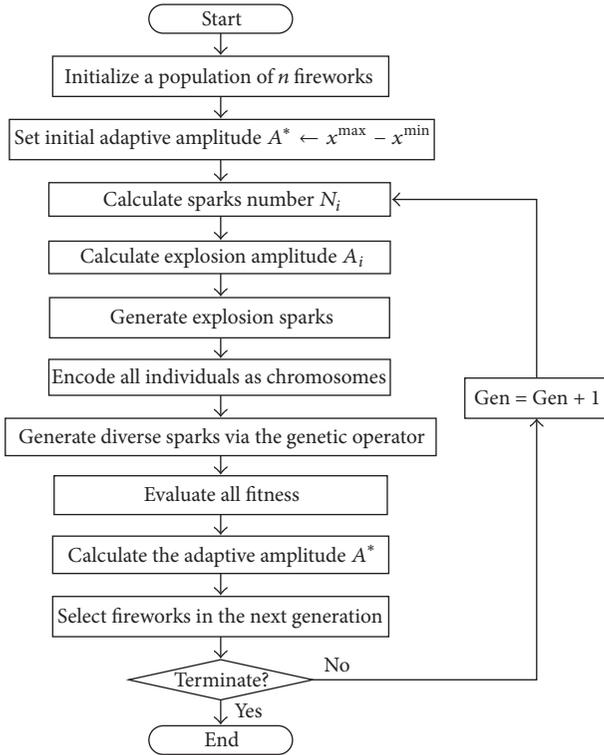


FIGURE 7: The flowchart of hybrid FWA.

- (a) With the stochastic robustness indices listed in Table 1 and the indicator function $I[\cdot]$ as defined in (19), calculate the indicator function value for the corresponding index.
- (b) By the Monte Carlo simulation, N samples under uncertainties are generated to estimate

the probability $\hat{P}(d)$ in which the design requirements of HV control are not satisfied.

- (c) For all the solutions generated in the search process, calculate the cost function $\hat{J}(d)$ in (20).

- (3) Prepare for the next step searching.

- (a) After the evaluation of all solution's fitness, the optimal solution is selected as a firework in the next generation. Its adaptive amplitude A^* is calculated based on (28).

- (b) Randomly select $n - 1$ fireworks among all the individuals.

- (4) Check if the stop criterion is fulfilled.

- (5) The optimal HOSM parameters are obtained, and an excellent HV tracking performance under uncertainties is achieved.

5. Simulation Study

5.1. Computational Intelligence Algorithm Test Cases. In this section, typical nonlinear benchmark functions in [25] are employed to test the effectiveness of the proposed hybrid FWA. For the comparison, the GA, PSO, AFWA, and proposed hybrid FWA are run on the benchmarks for 300000 evaluations per function. Each experiment for testing algorithm is repeated 50 times.

In the testing, the parameters settings of algorithms are listed in Table 2.

The first function is the Bent Cigar function and is described as follows:

$$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^D x_i^2, \quad (29)$$

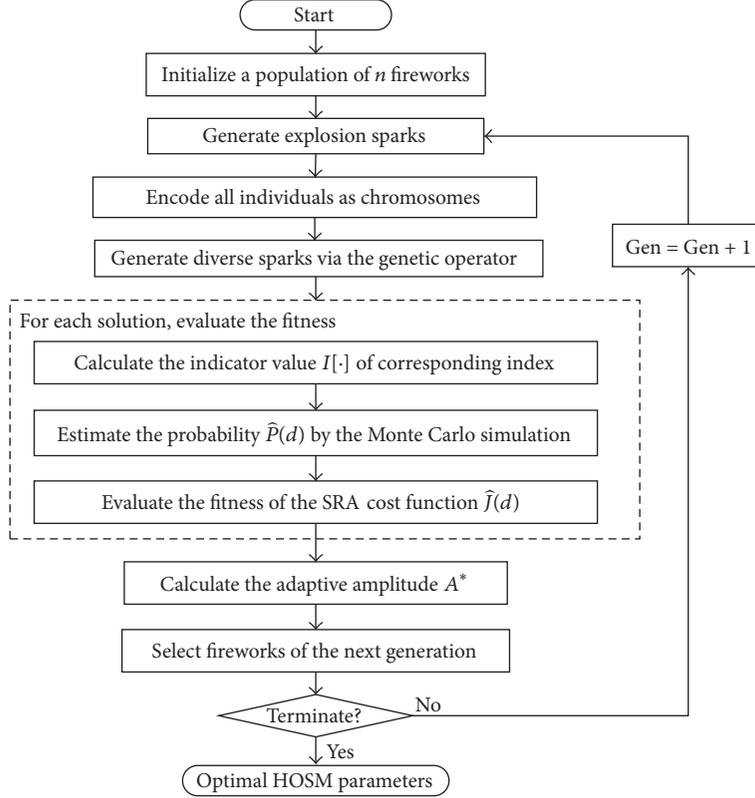


FIGURE 8: Flowchart of the hybrid FWA-based parameter optimization method.

TABLE 2: Parameters settings in algorithms.

Algorithms	Algorithm coefficients
GA	Population size: 200. Binary coded chromosome length: 10. Crossover probability: 0.7. Mutation probability: 0.015.
PSO	Particle number: 30. Inertia weight: $\omega_{\max} = 0.9$ $\omega_{\min} = 0.4$. Learning factor: $c_1 = c_2 = 1.5$.
AFWA	Total sparks number: 200. Other parameters are the same as in [20].
Hybrid FWA	Genetic operator parameters: $m = 15$, $P_r = 0.5$, $P_c = 0.95$, and $P_m = 0.025$. Other parameters are the same as AFWA.

where $x_i \in [-100, 100]$, $D = 4$. The Bent Cigar function is a unimodal function and is smooth. However, it has a narrow ridge. It has the global minimum $f_1^* = 0$ when $x_i = 0$, $i = 1, \dots, D$. The second function is the Rosenbrock function that is described as follows:

$$f_2(x) = \sum_{i=1}^{D-1} \left(100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right), \quad (30)$$

where $x_i \in [-100, 100]$, $D = 30$. The Rosenbrock function is a nonconvex function in which the global minimum is inside a long, narrow, and parabolic shaped flat valley. It has the global minimum $f_2^* = 0$ when $x_i = 1$, $i = 1, \dots, D$. The third function is the Griewank function described as follows:

$$f_3(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (31)$$

where $x_i \in [-100, 100]$, $D = 30$. The Griewank function is a multimodal function. It has the global minimum $f_3^* = 0$ when $x_i = 0$, $i = 1, \dots, D$. The fourth function is the Alpine function described as follows:

$$f_4(x) = \sum_{i=1}^D \left| \frac{1}{10} x_i \sin\left(\frac{x_i}{10}\right) + x_i \right|, \quad (32)$$

where $x_i \in [-100, 100]$, $D = 30$. The Alpine function is a multimodal function. It has the global minimum $f_4^* = 0$ when $x_i = 0$, $i = 1, \dots, D$. The fifth function is the Rastrigin function that is described in

$$f_5(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad (33)$$

where $x_i \in [-100, 100]$, $D = 30$. The Rastrigin function is a multimodal function, which has huge number of local

TABLE 3: Algorithm testing results.

Function ID	Metric	GA	PSO	AFWA	Hybrid FWA
1	Mean	2.0817e + 4	63.8216	44.4774	36.6983
	Std.	1.3336e + 4	59.7943	40.1607	39.9258
2	Mean	234.6298	628.8852	58.3773	56.1416
	Std.	123.0631	477.8930	31.7385	23.7506
3	Mean	1.2007	1.7338	0.7527	0.7125
	Std.	0.2004	0.5626	0.2188	0.1904
4	Mean	0.2287	0.6958	0.0708	0.0658
	Std.	0.0687	0.4486	0.0259	0.0250
5	Mean	22.9280	55.2201	19.6178	16.8546
	Std.	24.7651	24.7651	15.2610	12.0650
6	Mean	4.8939	3.7720	3.8627	3.6126
	Std.	0.9717	1.2440	1.0428	0.8973

optima. It has the global minimum $f_5^* = 0$ when $x_i = 0$, $i = 1, \dots, D$. The last function is the expanded Schaffer F6 function described in

$$g(x, y) = 0.5 + \frac{\left(\sin^2\left(\sqrt{x^2 + y^2}\right) - 0.5\right)}{\left(1 + 0.001(x^2 + y^2)\right)^2}, \quad (34)$$

$$f_6(x) = g(x_1, x_2) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1),$$

where $x_i \in [-100, 100]$, $D = 30$. The expanded Schaffer F6 function is a multimodal function. It has the global minimum $f_6^* = 0$ when $x_i = 0$, $i = 1, \dots, D$.

The testing results are given in Table 3.

As shown in Table 3, the proposed hybrid FWA presents the means closest to global minimum. Therefore, the testing results indicate that the hybrid FWA proposed in this study exhibits better search efficiency, when compared to the GA, PSO, and AFWA.

5.2. Algorithm Analysis in Parameter Optimization. In order to analyze the parameter searching efficiency of algorithms, the GA, PSO, AFWA, and proposed hybrid FWA are used to search for the optimal design parameters of the HOSM controller of HV. In the search, for all the algorithms, the number of individuals is 32, and the number of iterations is 15. For the AFWA, the number of fireworks is 5, the total number of sparks is 32, and the number of mutation sparks is 4. For the hybrid FWA, the number of fireworks and total sparks is the same as in AFWA, $m = 3$, and $P_m = 0.1$. The other parameters of algorithms are set the same as shown in Table 2.

The ranges of the uncertainties in HV are as follows:

$$|\Delta m| \leq 0.5\%,$$

$$|\Delta \rho| \leq 10\%,$$

$$|\Delta C_L| \leq 10\%,$$

$$|\Delta C_D| \leq 10\%,$$

TABLE 4: Search space of HV controller parameters.

Controller parameters	Bound
$\alpha_{\xi v}$	[5, 50]
$\beta_{\xi v1}$	[0.01, 10]
$\beta_{\xi v2}$	[0.01, 10]
$\alpha_{\xi h}$	[10, 100]
$\beta_{\xi h1}$	[0.01, 10]
$\beta_{\xi h2}$	[0.01, 10]
$\beta_{\xi h3}$	[0.01, 10]

$$|\Delta C_T| \leq 10\%,$$

$$|\Delta C_M| \leq 10\%.$$

(35)

The search space of the HV controller parameters is given in Table 4.

As given in Table 1, the cost function $J(d)$ in SRA is a weighted sum of 10 probabilities of the design requirements to guide the search of the HOSM controller parameters. The parameters specified for the indicator function $I[\cdot]$ are as follows: $TTP_{h,1} = 5$, $TTP_{h,2} = 1$, $TTP_{v,1} = 0.05$, $TTP_{v,2} = 0.02$, $STP_{h,1} = 3$, $STP_{h,2} = 1$, $STP_{v,1} = 0.05$, $STP_{v,2} = 0.005$, $FCP_{h,1} = 1$, $FCP_{h,2} = 0.2$, $FCP_{v,1} = 1$, $FCP_{v,2} = 0.2$, $CE_{h,1} = 2$, $CE_{h,2} = 1$, $CE_{v,1} = 2$, and $CE_{v,2} = 1$. The duration of interested transient stage is defined by the parameters $a_h = 0.15$ and $a_v = 0.4$.

The results of the HV performance optimization using various optimization algorithms are shown in Figure 9. The x-axis of the figure shows the number of iterations, and the y-axis shows the optimal value of the cost function $\hat{J}(d)$. The comparative result indicates that the proposed hybrid FWA exhibits better global search ability for the optimal HV control parameters than that of the GA, PSO, and AFWA.

5.3. Results of Optimal HOSM Controller Design. With the proposed hybrid FWA-based parameter optimization method, we shall examine the performance of the optimal HOSM controller in the trajectory tracking of HV. Initially,

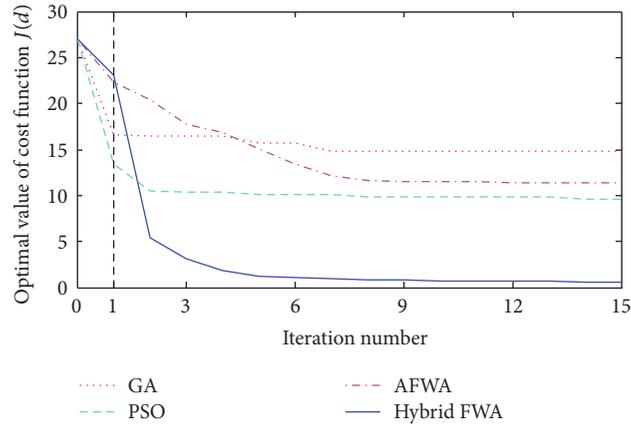


FIGURE 9: HV performance optimization using various algorithms.

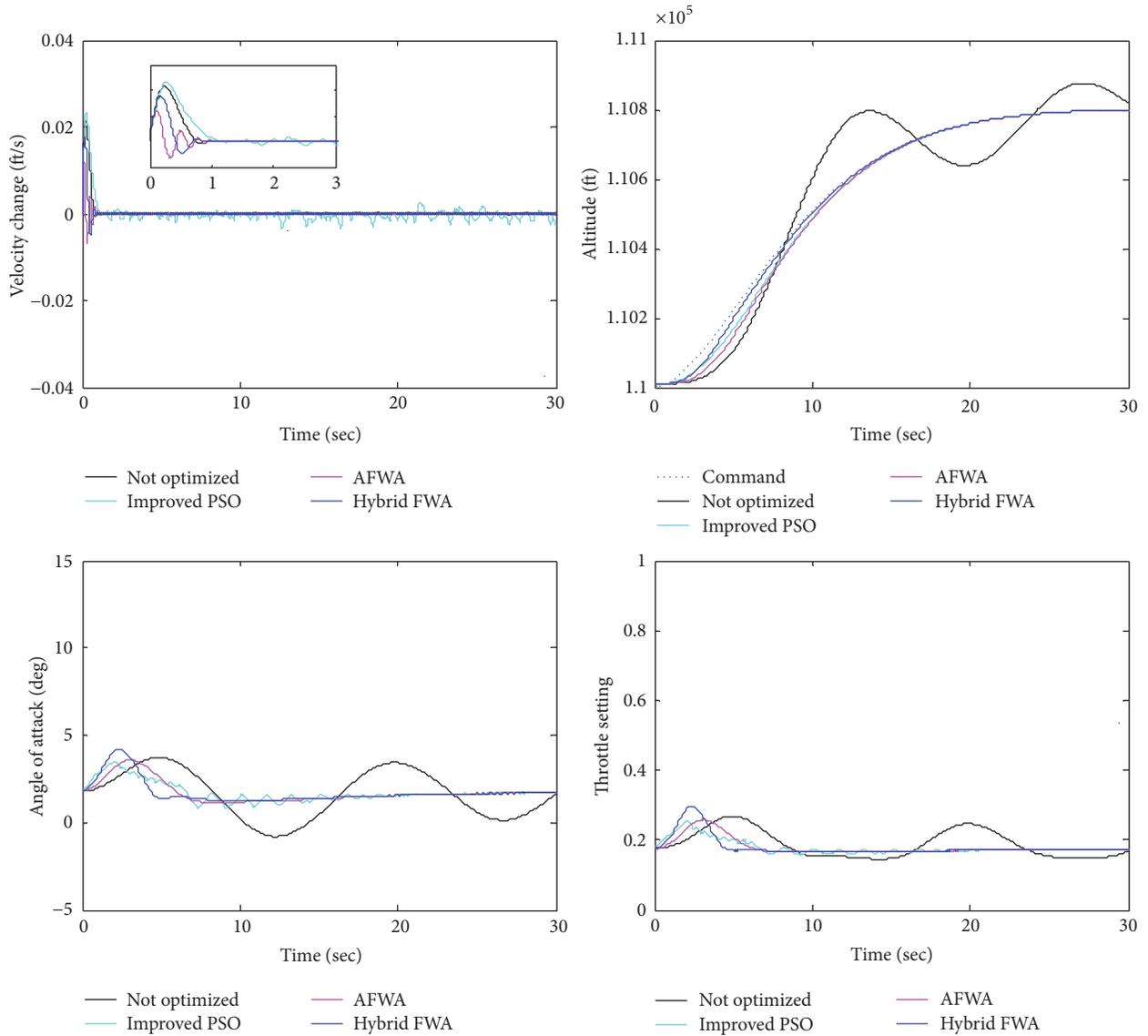


FIGURE 10: Tracking trajectories using various controller parameters.

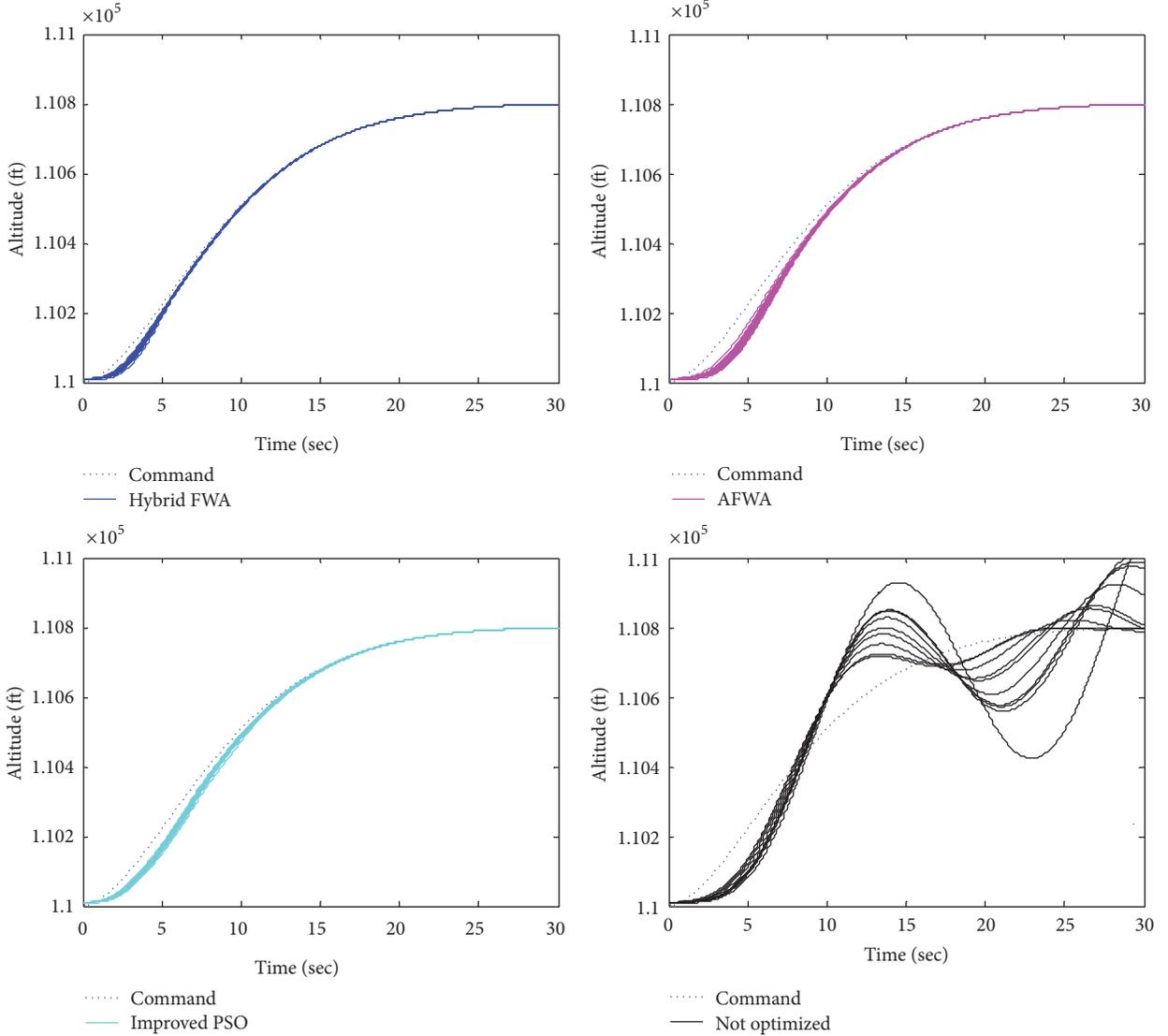


FIGURE 11: Random responses using various controller parameters.

the cruising flight conditions are as follows: Mach number $M_a = 15$, $V = 15060$ ft/s, $h = 110000$ ft, $\gamma = 0$ deg, and $q = 0$ deg/s. At the cruising flight conditions, the aerodynamic parameters C_L , C_D , C_T , $C_M(\alpha)$, $C_M(\alpha, \delta_e)$, and $C_M(\alpha, q)$ are given as follows:

$$\begin{aligned}
 C_L &= 0.6203\alpha, \\
 C_D &= 0.6450\alpha^2 + 0.0043378\alpha + 0.003772, \\
 C_T &= \begin{cases} 0.02576\beta, & \beta \leq 1, \\ 0.0224(1 + 0.15\beta), & \beta > 1, \end{cases} \\
 C_M(\alpha) &= -0.035\alpha^2 + 0.036617\alpha + 5.3261 \times 10^{-6},
 \end{aligned}$$

$$C_M(\alpha, \delta_e) = c_e(\delta_e - \alpha), \quad c_e = 0.0292,$$

$$C_M(\alpha, q) = \frac{\bar{c}q}{2V} (-6.796\alpha^2 + 0.3015\alpha - 0.2289).$$

(36)

After 15 search iterations by the proposed hybrid FWA-based parameter optimization algorithm, the optimal quasi-continuous HOSM controller parameters are determined as follows: $d_H^* = \{28.50, 1.74, 2.90, 56.66, 0.39, 1.22, 6.85\}$. Using AFWA, the optimal controller parameters are determined as follows: $d_A^* = \{40.65, 1.36, 8, 35, 0.40, 1.15, 4\}$. For comparison purposes, the other two sets of design parameters are given: The quasi-continuous HOSM controller parameters (not optimized) in [8] are as follows: $d_N = \{30, 1, 2, 10, 0.5, 1, 3\}$. The HOSM controller parameters

determined by the improved PSO in [11] are as follows: $d_P^* = \{107.33, 0.67, 2.83, 22.29, 125.38, 0.54, 1.06, 5.50, 26.64\}$.

In order to demonstrate the tracking performance of HV under uncertainties, the command tracking trajectories using four sets of controller parameters are given in Figure 10. In the simulation, the reference command is generated to control the HV to climb 800 ft at constant velocity in about 15 s. The parametric uncertainties are set as follows: $\Delta m = 0.1\%$, $\Delta \rho = 2\%$, $\Delta C_L = 2\%$, $\Delta C_D = 2\%$, $\Delta C_T = 2\%$, and $\Delta C_M = 2\%$, which are within the range given in (35).

In Figure 10, the trajectories of altitude h , velocity V , angle of attack α , and throttle setting β are depicted by the solid lines, and the reference command is shown by the dotted line. The simulation results demonstrate that the optimal controller parameters d_H^* determined by hybrid FWA provide a stable and high-accuracy tracking of the reference command in the presence of uncertainties. The command tracking error of the HV control system using the parameters d_H^* remains the smallest, when compared to the controller parameters d_A^* , d_P^* , and d_N . Besides, a faster dynamic response as well as lower fuel consumption is achieved using the parameters determined by the proposed hybrid FWA.

Next, with randomly generated uncertainties, the command tracking trajectories using four sets of controller parameters are demonstrated in Figure 11. The uncertain parameters are assumed to be uniformly distributed within the bounds given in (35). The results indicate that the optimal controller parameters d_H^* determined by the proposed hybrid FWA not only guarantee the HV system stability, but also exhibit a better tracking performance under bounded uncertainties.

Therefore, the simulation results demonstrate that the HV controller designed by the proposed hybrid FWA-based parameter optimization method achieves an excellent tracking performance in the presence of uncertainties.

6. Conclusion

In this study, we propose an improved hybrid FWA-based parameter optimization method for nonlinear HV control under uncertainties. An approach of searching for the optimal design parameters is developed by using two processes. The first process involves modeling the relation between the design parameters and the cost function that evaluates the likelihood of system instability and design requirement violation by using SRA. Subsequently, the cost function is minimized by the improved hybrid FWA to achieve a satisfactory tracking performance for the HV system with uncertainties. The proposed method makes it easier and more efficient to solve the optimization problem of satisfying all the HV design requirements with high probability. When compared with other algorithms, the hybrid FWA exhibits better efficiency in solving the HV parameter optimization problem with respect to uncertainties. Moreover, it is also efficient in solving other complex optimization problems.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China (Grant nos. 61573161 and 61473124).

References

- [1] B. Xu, D. Wang, Y. Zhang, and Z. Shi, "DOB based neural control of flexible hypersonic flight vehicle considering wind effects," *IEEE Transactions on Industrial Electronics*, vol. PP, no. 99, p. 1, 2017.
- [2] Y. Chang, T. Jiang, and Z. Pu, "Adaptive control of hypersonic vehicles based on characteristic models with fuzzy neural network estimators," *Aerospace Science and Technology*, vol. 68, pp. 475–485, 2017.
- [3] J. Wang, Y. Wu, and X. Dong, "Recursive terminal sliding mode control for hypersonic flight vehicle with sliding mode disturbance observer," *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1489–1510, 2015.
- [4] H. An, C. Wang, and B. Fidan, "Sliding mode disturbance observer-enhanced adaptive control for the air-breathing hypersonic flight vehicle," *Acta Astronautica*, vol. 139, pp. 111–121, 2017.
- [5] Y.-J. Wu, J.-X. Zuo, and L.-H. Sun, "Adaptive terminal sliding mode control for hypersonic flight vehicles with strictly lower convex function based nonlinear disturbance observer," *ISA Transactions*, 2017.
- [6] A. Levant, "Quasi-continuous high-order sliding-mode controllers," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 50, no. 11, pp. 1812–1816, 2005.
- [7] M. Sagliano, E. Mooij, and S. Theil, "Adaptive disturbance-based high-order sliding-mode control for hypersonic-entry vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 3, pp. 521–536, 2017.
- [8] Y. Zhang, R. Li, T. Xue, Z. Liu, and Z. Yao, "An analysis of the stability and chattering reduction of high-order sliding mode tracking control for a hypersonic vehicle," *Information Sciences*, vol. 348, pp. 25–48, 2016.
- [9] R. F. Stengel and L. R. Ray, "Stochastic robustness of linear time-invariant control systems," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 36, no. 1, pp. 82–87, 1991.
- [10] Q. Wang and R. F. Stengel, "Robust nonlinear control of a hypersonic aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 4, pp. 577–585, 2000.
- [11] L. Cao, D. Zhang, S. Tang, and F. Deng, "A practical parameter determination strategy based on improved hybrid PSO algorithm for higher-order sliding mode control of air-breathing hypersonic vehicles," *Aerospace Science and Technology*, vol. 59, pp. 1–10, 2016.
- [12] Q. Wang and R. Stengel, "Robust nonlinear control of a hypersonic aircraft," in *Proceedings of the Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Portland, OR, USA, 1999.

- [13] Q. Wang and R. F. Stengel, "Robust nonlinear flight control of a high-performance aircraft," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 1, pp. 15–26, 2005.
- [14] A. Azizi, "Introducing a novel hybrid artificial intelligence algorithm to optimize network of industrial applications in modern manufacturing," *Complexity*, vol. 2017, Article ID 8728209, 2017.
- [15] Y. Li, Y. Wu, and X. Qu, "Chicken Swarm-Based Method for Ascent Trajectory Optimization of Hypersonic Vehicles," *Journal of Aerospace Engineering*, vol. 30, no. 5, Article ID 04017043, 2017.
- [16] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning," *Choice Reviews Online*, vol. 27, no. 02, pp. 27-0936–27-0936, 1989.
- [17] A. Taieb, M. Soltani, and A. Chaari, "Parameter Optimization of MIMO Fuzzy Optimal Model Predictive Control By APSO," *Complexity*, vol. 2017, Article ID 5813192, 11 pages, 2017.
- [18] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12–15, 2010, Proceedings, Part I*, vol. 6145 of *Lecture Notes in Computer Science*, pp. 355–364, Springer, Berlin, Germany, 2010.
- [19] S. Bureerat, "Hybrid population-based incremental learning using real codes in," in *Proceedings of the 5th international conference on Learning and Intelligent Optimization*, pp. 379–391, Springer-Verlag, Rome, Italy, 2011.
- [20] J. Li, S. Zheng, and Y. Tan, "Adaptive fireworks algorithm," in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 3214–3221, China, July 2014.
- [21] Y.-J. Zheng, X.-L. Xu, H.-F. Ling, and S.-Y. Chen, "A hybrid fireworks optimization method with differential evolution operators," *Neurocomputing*, vol. 148, pp. 75–82, 2015.
- [22] B. Zhang, Y.-J. Zheng, M.-X. Zhang, and S.-Y. Chen, "Fireworks Algorithm with Enhanced Fireworks Interaction," *IEEE Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 42–55, 2017.
- [23] J. Li, S. Zheng, and Y. Tan, "The Effect of Information Utilization: Introducing a Novel Guiding Spark in the Fireworks Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 153–166, 2017.
- [24] J. T. Parker, A. Serrani, S. Yurkovich, M. A. Bolender, and D. B. Doman, "Control-oriented modeling of an air-breathing hypersonic vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 3, pp. 856–869, 2007.
- [25] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, 2013."

Research Article

An Improved Test Selection Optimization Model Based on Fault Ambiguity Group Isolation and Chaotic Discrete PSO

Xiaofeng Lv,^{1,2} Deyun Zhou,¹ Yongchuan Tang ,¹ and Ling Ma²

¹School of Electronics and Information, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

²Naval Aviation University, Yantai, Shandong 264001, China

Correspondence should be addressed to Yongchuan Tang; tangyongchuan@mail.nwpu.edu.cn

Received 24 September 2017; Revised 22 November 2017; Accepted 28 November 2017; Published 15 January 2018

Academic Editor: László T. Kóczy

Copyright © 2018 Xiaofeng Lv 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.

Sensor data-based test selection optimization is the basis for designing a test work, which ensures that the system is tested under the constraint of the conventional indexes such as fault detection rate (FDR) and fault isolation rate (FIR). From the perspective of equipment maintenance support, the ambiguity isolation has a significant effect on the result of test selection. In this paper, an improved test selection optimization model is proposed by considering the ambiguity degree of fault isolation. In the new model, the fault test dependency matrix is adopted to model the correlation between the system fault and the test group. The objective function of the proposed model is minimizing the test cost with the constraint of FDR and FIR. The improved chaotic discrete particle swarm optimization (PSO) algorithm is adopted to solve the improved test selection optimization model. The new test selection optimization model is more consistent with real complicated engineering systems. The experimental result verifies the effectiveness of the proposed method.

1. Introduction

High technologies contribute a lot to the improvement of the performance of complex equipment. However, new challenges are also brought into testing, diagnosis, and maintenance stages of the equipment because the system structure is more and more complicated [1–3]. In order to improve the testing efficiency and diagnosis ability, as well as to reduce the test cost, the design for testability (DFT) should be carried out simultaneously during the design of the equipment [4]. Test optimization selection is a process of selecting the proper test set from all the available test set under the constraints including minimum test cost, test cycle, fault detection rate (FDR) index, fault isolation rate (FIR) index, test resources, and engineering-oriented constraint rules [5]. Test optimization selection is a nondeterministic polynomial-time hard (NP-hard) problem [6], which is a challenging problem related to the combination problem and the permutation problem.

In general, two categories of methods have been applied to solve the test optimization selection problem: sequential

fault diagnosis methods [7–11] and direct searching algorithms [12–15]. Sequential fault diagnosis methods include dynamic programming (DP) algorithm, information entropy method, graph search method, and heuristics method. These methods model the test optimization selection problem as a binary identification problem [7, 16]. Direct searching algorithms are intelligent optimization algorithms, including evolution algorithms and particle swarm optimization (PSO) algorithm. The direct searching method with artificial intelligence searching method can directly find the optimal test set, for example, the multidimensional discrete PSO algorithm [17, 18], the genetic algorithm (GA) [13], and so on [19]. With these methods, although the optimal solution can be found, some faults with low probability cannot be detected and isolated, which means information loss for the system.

In the process of test selection optimization, among the alternative test sets, if the test cost of each test is equal to each other, the test set corresponding to faults of high fault rate is usually chosen. This may lead to failure in fault detection and isolation of those faults with low probability. More seriously for the complex system, the losing of some fault information

may lead to increasing of the maintenance costs of equipment system. In addition, if all those faults with low probability are taken into consideration by increasing the FDR, the test cost will increase significantly. In real engineering practice, it is unreasonable and uneconomical to realize the unique isolation of all the faults. Thus a process of fault detection and isolation that can isolate the fault into a specific replacement unit or some fault ambiguity groups is consistent with real applications. In this paper, an improved test selection optimization model is proposed. Firstly, a fault is isolated to a fault ambiguity group with a number of L replacement unit (in real applications, L is usually chosen to be 3). After that, the maintenance methods such as replacement are adopted to isolate the fault to a specific replacement unit. The proposed strategy is different from the methods in [20, 21], where the fault ambiguity degree of FIR is 1. A more flexible model for test selection optimization model will be proposed in this paper, which can be more compatible with real engineering applications.

Computational intelligence (CI) theories such as evolutionary algorithms [22, 23], artificial neural networks [24], cognitive map analysis [25], Physarum solver [26–28], fuzzy sets [29–31], belief function [32–34], PSO [35–37], and so on [38], have been widely used to cope the complex problems including the permutation flow shop problem [39], supply chain network [40, 41], traveling salesman problem [42], pattern recognition [43–46], power system [47], product design and manufacturing [48], and so on [49–52]. Recently, based on this progress in CI, many nature inspired approaches have been proposed to solve test selection optimization problem, such as the greedy strategy [53], the genetic algorithm [54, 55], the evolutionary algorithm [56, 57], and so on [58]. The PSO has some advantages in comparison with some other methods; for example, it requires less parameters and has a fast convergence rate. In this paper, the improved chaotic discrete PSO (ICDPSO) algorithm is applied to deal with the improved test selection optimization model. The indexes including FDR and FIR are handled simultaneously to solve the improved test selection optimization model. While modelling the fitness function, the fault ambiguity group with different ambiguity degree is taken into consideration. The reasonableness of the improved test selection optimization model as well as the effectiveness of the ICDPSO algorithm is verified according to the experiment. In addition, the computational performance of the proposed method is better than that of GA.

The rest of this paper is organized as follows. The preliminaries are introduced in Section 2. In Section 3, a new evidential sensor fusion method is proposed. The method of solving the improved test selection optimization model is introduced in Section 4. An application example in fault diagnosis is illustrated to show the efficiency of this method in Section 5. Finally, the conclusion is presented in Section 6.

2. Preliminaries

2.1. Fault Test Dependency Matrix. Fault test dependency matrix is a Boolean matrix describing the correlation between the system fault and test. Assume that the fault set is $F = \{f_1, \dots, f_m\}$, the alternative test set is $T = \{t_1, \dots, t_n\}$, and

the dependency matrix is a Boolean matrix with $m \times n$ dimensions, denoted as follows [59, 60]:

$$FT_{m \times n} = \begin{bmatrix} ft_{11} & ft_{12} & \cdots & ft_{1n} \\ ft_{21} & ft_{22} & \cdots & ft_{2n} \\ \vdots & \vdots & & \vdots \\ ft_{m1} & ft_{m2} & \cdots & ft_{mn} \end{bmatrix}, \quad (1)$$

where ft_{ij} is generally defined as a Boolean variable, which represents the correlation between the fault mode f_i and the test t_j .

If a fault mode f_i can be detected by the test t_j , then $ft_{ij} = 1$; otherwise $ft_{ij} = 0$. In the matrix, the vector $F_i = [ft_{i1}, ft_{i2}, \dots, ft_{in}]$ in the i th row represents the correlation of the fault mode f_i with all the tests. F_i is called the symptom of the fault mode f_i . The vector $T_j = [ft_{1j}, ft_{2j}, \dots, ft_{mj}]^T$ in the j th column represents the correlation of the test t_j with all the fault modes. T_j is called the symptom of the test t_j .

A test set that can be affected by the fault mode f_i is defined as $T(f_i) = \{t_j \mid ft_{ij} = 1, \forall t_j\}$. Similarly, a fault set that can be detected by the test t_j is defined as $F(t_j) = \{f_i \mid ft_{ij} = 1, \forall f_i\}$.

2.2. Testability Index. Testability index includes fault detection rate (FDR) and fault isolation rate (FIR).

2.2.1. Fault Detection Rate. Fault detection rate (FDR) is generally defined as, within the prescribed time, the ratio of the total fault rate of the fault modes to the total fault rate of the units under test. The condition of a fault mode f_i which can be detected by the test set T_s is that, in the fault test dependency matrix, at least one element of the row vector corresponding to the f_i is 1. Mathematically, the condition can be denoted as follows:

$$\bigcup_{t_j \in T_s}^{N_s} ft_{ij} = 1, \quad (2)$$

where \bigcup is the OR operation of Boolean variable and N_s is the element number in the set T_s . Then, all the fault modes detected by T_s can be denoted as a set F_D , shown as follows:

$$F_D = \left\{ f_i \mid f_i \in F, \bigcup_{t_j \in T_s}^{N_s} ft_{ij} = 1 \right\}. \quad (3)$$

Thus, FDR is defined as follows:

$$\gamma_{FD} = \frac{\sum_{f_i \in F_D} \lambda_i}{\sum_{f_i \in F} \lambda_i}, \quad (4)$$

where λ_i is the fault rate of the i th fault mode.

2.2.2. Fault Isolation Rate. Fault isolation rate (FIR) is generally defined as, within the prescribed time, with the specified method, and under the constraint that the number of the

replacement unit is no more than the requested replacement number L , the ratio of the total fault rate that can be isolated correctly to the detected total fault rate. The replacement number L represents the ambiguity of fault isolation. If $L = 1$, it is called the unique isolation. If $L > 1$, it is called ambiguity isolation.

Assume that $T(f_i)$ and $T(f_j)$ are the symptom sets of the fault modes f_i and f_j , respectively; then the condition that a fault mode f_i can be isolated can be shown as follows:

$$T(f_i) \oplus T(f_j) = 1, \quad \forall f_j \in F_D, i \neq j, \quad (5)$$

where \oplus is the exclusive OR operator of two sets. If two sets are the same, the calculation result with \oplus is 0; otherwise, the result is 1. Define an operator for two sets, denoted as \otimes , satisfying the following condition: if $T(f_i) \oplus T(f_j) = 0$, then $T(f_i) \otimes T(f_j) = 1$; that is, the fault modes f_i and f_j belong to the same fault ambiguity group.

For a given ambiguity of fault isolation L , the fault set, denoted as F_I , that can be isolated by a test set T_s is defined as follows:

$$F_I = \left\{ f_i \mid f_i \in F_D, \sum_{f_j \in F_D} T(f_i) \otimes T(f_j) \leq L, \forall f_j \in F_D, i \neq j \right\}. \quad (6)$$

FIR is defined as follows:

$$\gamma_{FI} = \frac{\lambda_L}{\lambda_D} = \frac{\sum_{f_i \in F_I} \lambda_i}{\sum_{f_i \in F_D} \lambda_i}, \quad (7)$$

where λ_D represents the total fault ratio of all the detected fault modes and λ_L represents the total fault ratio of the fault modes that can be isolated under the constraint that the number of the replaced units is no more than the replacement number L .

3. The Improved Test Selection Optimization Model

In real applications, the value of ambiguity of fault isolation (L) is generally not bigger than 3. Without lose generality, we define FIR as follows:

$$\gamma_{FIL} = \frac{\sum \lambda_{1i} + \sum \lambda_{2j} + \sum \lambda_{3k} + \cdots + \sum \lambda_{Lp}}{\lambda_D}, \quad (8)$$

where L represents the ambiguity of fault isolation and $\sum \lambda_{Lp}$ is the fault rate of the p th fault mode with the isolated replacement unit number of the fault being L . Equation (8) contains the following conditions:

- (1) $L = 1$: in this case, $\gamma_{FI1} = \sum \lambda_{1i} / \lambda_D$, where λ_{1i} is the fault rate of the i th fault mode while the isolated replacement unit number of the fault is 1.

- (2) $L = 2$: in this case, $\gamma_{FI2} = (\sum \lambda_{1i} + \sum \lambda_{2j}) / \lambda_D$, where λ_{2j} is the fault rate of the j th fault mode while the isolated replacement unit number of the fault is 2.

- (3) $L = 3$: in this case, $\gamma_{FI3} = (\sum \lambda_{1i} + \sum \lambda_{2j} + \sum \lambda_{3k}) / \lambda_D$, where λ_{3k} is the fault rate of the k th fault mode while the isolated replacement unit number of the fault is 3.

The aim of the improved test selection optimization model is selecting the optimized test set from the alternative test sets of the system to meet the requirements of the system testability index as well as to minimize the test cost corresponding to the selected test set. Assume that $\lambda = [\lambda_1, \dots, \lambda_m]^T$ is the vector of prior probability of fault modes, $C = [c_1, \dots, c_n]$ is the vector of test cost, and the selected test set is T_s , where $T_s \subseteq T$. The total test cost of the test set T_s is defined as follows:

$$C_{T_s} = \sum_{t_j \in T_s} c_j. \quad (9)$$

Based on the constraint of testability index and the objective function, the proposed test selection optimization model is defined as follows:

$$\begin{aligned} \min \quad & \{C_{T_s}\}, \\ \text{s.t.} \quad & \gamma_{FD} \geq \gamma_{FD}^*, \\ & \gamma_{FI1} \geq \gamma_{FI1}^*, \\ & \gamma_{FI2} \geq \gamma_{FI2}^*, \\ & \gamma_{FI3} \geq \gamma_{FI3}^*, \\ & \vdots \\ & \gamma_{FIL} \geq \gamma_{FIL}^*, \end{aligned} \quad (10)$$

where γ_{FD}^* , γ_{FI1}^* , γ_{FI2}^* , γ_{FI3}^* , and γ_{FIL}^* are the desired fault detection rate of the system, the fault isolation rate with an ambiguity degree 1, the fault isolation rate with an ambiguity degree 2, the fault isolation rate with an ambiguity degree 3, and the fault isolation rate with an ambiguity degree L , respectively. The fault isolation rate of the ambiguity degree L enables the proposed test selection optimization model to be more flexible in real applications.

4. Solving the Improved Test Selection Optimization Model

The process of test selection optimization is a multiobjective optimization problem. The ICDPSO algorithm has been adopted to solve the improved test selection optimization model.

4.1. Chaos Initialization. The chaotic motion seems to be random, but it has a hidden exquisite structure such as the characteristics of ergodicity, randomness, and regularity. The basic idea of this algorithm is, firstly, to generate a set of chaotic variables, which has the same number as the

optimization variables; secondly, to introduce the chaos into the optimization variables with a way similar to carrier and to amplify the search scope of chaotic motion to the range of the value of the optimization variables simultaneously; finally, to search for and optimize the chaotic variables directly.

The improved Tent map has better ergodic uniformity and is beneficial to global optimization of chaos; thus, it is adopted to generate the chaotic variables. The Tent map is shown as follows:

$$z_{k+1} = T(z_k) = \begin{cases} 2z_k, & 0 \leq z_k \leq 0.5, \\ 2(1 - z_k), & 0.5 < z_k \leq 1. \end{cases} \quad (11)$$

The Tent map may iterate to a fixed point, for example, 0, 0.25, 0.5, 0.75; to avoid this situation, a small disturbance is added to the sequence to enable the Tent map to go back to chaos. The added disturbance is shown as follows:

$$\begin{aligned} & \text{if } Z_k = 0, 0.25, 0.5, 0.75 \text{ or } Z_k = Z_{k-m}, \\ & \text{then } Z_{k+1} = T(Z_k) + 0.1 \text{ rand}(0, 1), \\ & \text{else } Z_{k+1} = T(Z_k), \end{aligned}$$

where Z_k is a chaotic variable, $z_k \in [0, 1]$, $m = 1, 2, 3, 4, 5$. The improved Tent map enables the ergodic process of Z_k to cover the whole range among $[0, 1]$. The iterative results starting from an arbitrary initial value of $z_0 \in [0, 1]$ can be a deterministic chaotic time sequences: z_1, z_2, z_3 , etc.

4.2. Calculation of Fitness Function. The fitness value aims to evaluate the pros and cons of each particle in the group, of which the calculation method has an important influence on the performance of a particle swarm. To solve the test selection optimization model in (10), the FDR and the penalty function are adopted to define the fitness function; the fitness degree of the test set T_i is defined as follows:

$$\begin{aligned} \text{Fitness} = & \frac{\sum_{t_i \in T} c_i}{\sum_{t_i \in T_i} c_i + \sum_{t_i \in T} c_i} - \alpha \cdot \max(0, \gamma_{FI1}^* - \gamma_{FI1}) \\ & - \beta \cdot \max(0, \gamma_{FI2}^* - \gamma_{FI2}) - \theta \\ & \cdot \max(0, \gamma_{FI3}^* - \gamma_{FI3}) - \dots - \tau \\ & \cdot \max(0, \gamma_{FIL}^* - \gamma_{FIL}), \end{aligned} \quad (12)$$

where α, β, θ , and τ are penalty factors and they are constants, the penalty function aims to correct the particle that crossed the boundary, and the heuristic strategy is used to process the FDR to get the feasible particle.

According to the fitness degree defined in (12), if a particle in the group meet all the L requirements of FIR, its fitness is defined by the first term of (12); thus, the particle has a bigger fitness degree; if a particle in the group does not meet the 3 requirements of FIR, a penalty function is used to penalize the corresponding fitness degree; thus, its fitness degree is much more smaller than the first term of (12); in this way, the chance of this particle entering the next generation can be decreased.

4.3. Inertia Weight Adaptive Adjustment. In the discrete particle swarm algorithm, the purpose of the inertia weight ω is to control the effect of the former speed on the current speed, which is also for balancing the exploration and exploitation ability of the algorithm. After the initialization of the particle, in the early stage, the movement of the particle is not objective, the searching range is the whole solution space, and, thus, the value of ω should be bigger than the other cases. With the ongoing of the search, the searching range becomes smaller and will finally determine the range of search at the vicinity of the optimal solution; in this case, the value of ω should be small until the optimal solution is got. In this paper, the adjustment strategy is based on the degree of premature convergence of the particle, and the inertia weight of the particle is self-adaptively adjusted. The degree of premature convergence of the particle can be assessed with the following equation:

$$\Delta = |f_g - f_{ap}|, \quad (13)$$

where f_g is the fitness value of the global optimum particle and f_{ap} is the average fitness value of the particle whose fitness value is higher than the average fitness value of the current particle swarm (denoted as f_{ag}). The assessment rule is that the smaller the Δ is, the easier the particle swarm gets premature.

If the fitness value of a particle is f_i , the adjustment strategy is defined as follows:

- (1) If $f_i > f_{ap}$, which means that the corresponding particle has a good fitness in the whole population and it is near the vicinity of the optimal solution, then a small inertia weight should be assigned on the particle to enhance its local search ability. In this case, the adjustment strategy is defined as follows:

$$\omega(t) = \omega_s - (\omega_s - \omega_{\min}) \cdot \left| \frac{f_i - f_{ap}}{\Delta} \right|, \quad (14)$$

where ω_{\min} is chosen as the minimum of ω_s while doing parameter initialization and ω_s is the median of the value range of ω ; that is to say, $\omega_s = (\omega_{\max} + \omega_{\min})/2$.

- (2) If $f_{ag} < f_i < f_{ap}$, which means that the corresponding particle has a proper fitness in the whole population and it has a good global search ability as well as the local search ability, then the adjustment strategy is defined as follows:

$$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot \left(\frac{T-t}{T} \right)^2, \quad (15)$$

where T is the maximum iterations of the algorithm and t is the current number of iterations.

- (3) If $f_i < f_{ag}$, which means that the corresponding particle has bad fitness in the whole population and it is far from the vicinity of the optimal solution, then, a big inertia weight should be assigned on the particle

to enhance its global search ability. In this case, the adjustment strategy is defined as follows:

$$\omega(t) = \omega_{\max} - \frac{1}{1 + k_1 \cdot \exp(-k_2 \cdot \Delta)}, \quad (16)$$

where k_1 is the controlling parameter ($k_1 > 1$) and k_2 is the adjusting parameter ($k_2 < 0$).

4.4. Algorithm Implementation. The ICDPSO algorithm for test selection optimization is implemented as in the following steps.

Step 1 (parameter initialization). The initial value is assigned to the following parameters, respectively: the size of the population denoted as $Popsiz$, the learning factors denoted as c_1 and c_2 , the inertia weights denoted as ω_{\max} and ω_{\min} , the maximum iteration denoted as N_{\max} , and the constants such as α , β , θ , and τ .

Step 2 (generation of an initial population). The iteration counter of the population is set as $i = 0$. The initial population with a size M , which is denoted as $X(i) = \{x_1^i, x_2^i, \dots, x_M^i\}$, is generated based on chaos initialization.

Step 3 (calculation of fitness function). The fitness degree of each individual in the population $X(i)$ is calculated based on (12). Simultaneously, the individual extremum, denoted as P_{bestid} , and the global extremum, denoted as G_{bestid} , are updated.

Step 4 (inertia weight adjustment). The update of the population speed and location is based on the strategy of the inertia weight adaptive adjustment presented in Section 4.3. The next generation of population is $X(i + 1)$. The counter should be updated, which means $i \leftarrow i + 1$.

Step 5 (determination of the number of iteration). If the number of iterations has reached the maximum number, which is denoted as $i > N_{\max}$, then, jump to the next step (Step 6); else, go back to Step 3.

Step 6 (output). Output the results of test selection optimization.

The procedure of the algorithm solving the improved test selection optimization model is presented in Figure 1.

5. Application

The improved test selection optimization model is applied to a superheterodyne receiver system, which is adopted from [8]. There are 36 different alternative tests denoted as t_a ($a = 1, 2, 3, \dots, 36$) and 22 fault modes denoted as f_m ($m = 1, 2, 3, \dots, 22$) in the system. The fault probabilities, denoted as λ_i ($i = 1, 2, 3, \dots, 22$), of all the 22 fault modes are $10^{-3} \times [1.85, 9.23, 185, 1.85, 1.85, 9.23, 1.85, 9.23, 185, 185, 185, 1.85, 9.23, 185, 9.23, 1.85, 9.23, 1.85, 1.85, 1.85, 1.85, 1.85]$. The cost of each test is assumed to be a standard unit. The testability indicators for the system are shown in Table 1.

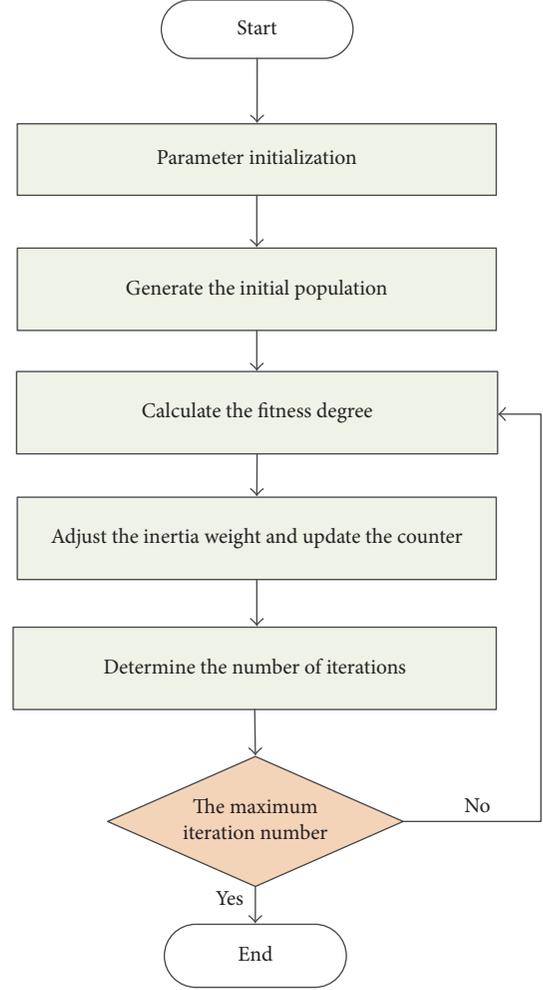


FIGURE 1: The procedure of the algorithm solving the proposed test selection optimization model.

TABLE 1: The testability indicators for the superheterodyne receiver.

Indicator	Minimum
FDR	95%
FIR with an ambiguity degree of 1	95%
FIR with an ambiguity degree of 2	98%
FIR with an ambiguity degree of 3	99%

The ICDPSO algorithm has been applied to solve the improved test selection optimization model of the superheterodyne receiver system. The values of the parameters are as follows: the size of the population is $Popsiz = 40$, the maximum iteration number is $N_{\max} = 200$, the inertia weights is $\omega_{\max} = 1.2$, the learning factors are $\omega_{\min} = 0.4$, $c_1 = 1.3$, $c_2 = 1.5$, and the penalty factors are $\alpha = \beta = \theta = 0.5$.

After 30 times of independent calculation with 200 iterations for each independent calculation, the fitness convergence curve of the particle can be got, as is shown in Figure 2. The optimal solutions for all the independent calculation can be got after about 45 to 95 iterations, which shows the convergence performance of the proposed method.

The optimal solutions of simulation are [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0], which means that the optimal complete test set is $[t_8, t_9, t_{26}, t_{28}, t_{31}, t_{32}, t_{34}]$. The corresponding fault set of the

optimal complete test set is $[f_1, f_2, f_3, f_5, f_6, f_8, f_9, f_{10}, f_{11}, f_{13}, f_{14}, f_{17}, f_{18}, f_{19}, f_{20}, f_{22}]$, as is shown in Table 2.

The FDR of the system is

$$\begin{aligned} \gamma_{FD} &= \frac{\sum_{f_i \in F_D} \lambda_i}{\sum_{f_i \in F} \lambda_i} \\ &= \frac{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5 + \lambda_6 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{13} + \lambda_{14} + \lambda_{17} + \lambda_{18} + \lambda_{19} + \lambda_{20} + \lambda_{22}}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{12} + \lambda_{13} + \lambda_{14} + \lambda_{15} + \lambda_{16} + \lambda_{17} + \lambda_{18} + \lambda_{19} + \lambda_{20} + \lambda_{21} + \lambda_{22}} \\ &= 0.9815. \end{aligned} \quad (17)$$

The fault set with an ambiguity degree of 1 is $[f_3, f_9, f_{10}, f_{11}, f_{14}, f_{17}, f_{20}]$; thus, the FIR with an ambiguity degree of 1 is

$$\gamma_{FI1} = \frac{\sum \lambda_{li}}{\lambda_D} = \frac{\lambda_3 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{14} + \lambda_{17} + \lambda_{20}}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5 + \lambda_6 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{13} + \lambda_{14} + \lambda_{17} + \lambda_{18} + \lambda_{19} + \lambda_{20} + \lambda_{22}} = 0.9530. \quad (18)$$

The fault set with an ambiguity degree of 2 is $[f_2, f_6, f_8, f_{18}, f_{19}, f_{22}]$; thus, the FIR with an ambiguity degree of 2 is

$$\begin{aligned} \gamma_{FI2} &= \frac{\sum \lambda_{li} + \sum \lambda_{2j}}{\lambda_D} = \frac{\lambda_2 + \lambda_3 + \lambda_6 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{14} + \lambda_{17} + \lambda_{18} + \lambda_{19} + \lambda_{20} + \lambda_{22}}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5 + \lambda_6 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{13} + \lambda_{14} + \lambda_{17} + \lambda_{18} + \lambda_{19} + \lambda_{20} + \lambda_{22}} \\ &= 0.9868. \end{aligned} \quad (19)$$

The fault set with an ambiguity degree of 3 is $[f_1, f_5, f_{13}]$; thus, the FIR with an ambiguity degree of 3 is

$$\begin{aligned} \gamma_{FI3} &= \frac{\sum \lambda_{li} + \sum \lambda_{2j} + \sum \lambda_{3k}}{\lambda_D} \\ &= \frac{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5 + \lambda_6 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{13} + \lambda_{14} + \lambda_{17} + \lambda_{18} + \lambda_{19} + \lambda_{20} + \lambda_{22}}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_5 + \lambda_6 + \lambda_8 + \lambda_9 + \lambda_{10} + \lambda_{11} + \lambda_{13} + \lambda_{14} + \lambda_{17} + \lambda_{18} + \lambda_{19} + \lambda_{20} + \lambda_{22}} = 1. \end{aligned} \quad (20)$$

The experimental result shows that (1) the FIRs with the ambiguity degrees of 1, 2, and 3 are 95.3%, 98.68%, and 100%, respectively; (2) the optimal complete test set is $[t_8, t_9, t_{26}, t_{28}, t_{31}, t_{32}, t_{34}]$, which means the test cost is 7; (3) there are 6 fault modes that cannot be detected and isolated by the optimal test set, including $f_4, f_7, f_{12}, f_{15}, f_{16}, f_{21}$. In comparison with the experimental results in [20], the number of fault modes that cannot be detected and isolated decreases from 9 to 6, which can be a marked improvement because 3 more fault modes in the system have been addressed. In addition, the FDR increases from 96.86% to 98.15%. This improvement should not be ignored especially for a core system or some

big systems, in which each tiny improvement may lead to a more safe or more economical system. The comparison of the experimental results between the proposed method and the method in [20] is shown in Table 3. In general, it is complicated to judge which method is better. The judgement should be dependent on the specific needs. However, two things are for sure: on the one hand, the effectiveness of the proposed method has been verified; on the other hand, the proposed method has a better performance than the method in [20], including a higher FDR and more detected fault modes. Last but not least, the proposed method can handle the FIR with a complicated ambiguity degree.

TABLE 2: The testability indicators for the superheterodyne receiver.

Fault mode	t_8	t_9	t_{26}	t_{28}	t_{31}	t_{32}	t_{34}	Fault probability ($\times 10^{-3}$)
f_1	0	1	0	0	0	0	0	1.85
f_2	1	1	0	1	1	1	1	9.23
f_3	0	0	1	0	0	0	0	185
f_4	0	0	0	0	0	0	0	1.85
f_5	0	1	0	0	0	0	0	1.85
f_6	1	1	0	1	1	1	1	9.23
f_7	0	0	0	0	0	0	0	1.85
f_8	0	1	0	0	0	1	0	9.23
f_9	0	1	0	1	0	1	0	185
f_{10}	0	1	0	0	1	0	0	185
f_{11}	0	1	0	0	0	0	1	185
f_{12}	0	0	0	0	0	0	0	1.85
f_{13}	0	1	0	0	0	0	0	9.23
f_{14}	0	1	0	1	1	1	1	185
f_{15}	0	0	0	0	0	0	0	9.23
f_{16}	0	0	0	0	0	0	0	1.85
f_{17}	1	1	0	0	0	0	1	9.23
f_{18}	0	1	0	0	0	1	0	1.85
f_{19}	1	1	1	1	1	1	1	1.85
f_{20}	0	1	0	0	1	1	0	1.85
f_{21}	0	0	0	0	0	0	0	1.85
f_{22}	1	1	1	1	1	1	1	1.85

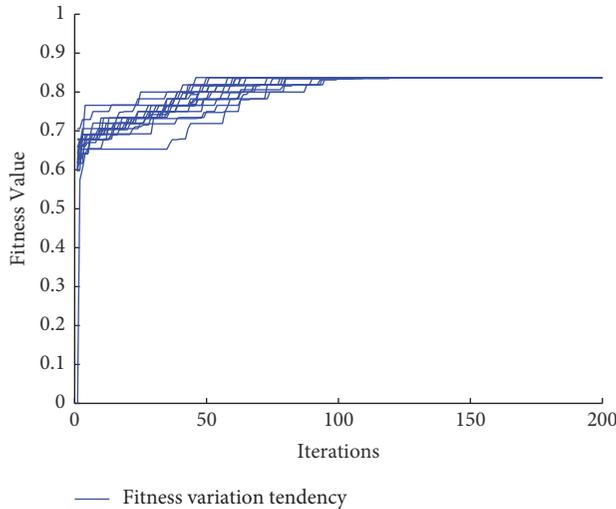


FIGURE 2: The fitness convergence curve of the superheterodyne receiver system.

Figure 3 presents the average convergence time of solving the test selection model in the application with ICDPSO algorithm, as well as the average convergence time with GA. In the aforementioned application, with GA, define that the population size is $Popsizel' = 40$, the maximum iteration number is $N'_{max} = 200$, the length of the chromosome is $Chrsize = 22$, the probabilities of crossover and mutation are $P_c = 0.7$ and $P_m = 0.01$, respectively, and the penalty factors are $\alpha' = \beta' = \theta' = 0.5$.

The curves of the fitness values with respect to the running time of ICDPSO and GA presented in Figure 3 show that, regarding the running time, ICDPSO has a significantly better performance in solving global optimization solution in comparison with GA. The running time of GA is about twice the cost of ICDPSO. After 30 times of independent experiments of the test selection optimization model for the superheterodyne receiver system, the statistical results of GA and ICDPSO are shown in Table 4. Although both ICDPSO and GA have the same minimum test cost of 7 and the success rates of searching out optimal solution are both 100%, the average running time of GA is more than 2 times that of ICDPSO. The priority of the proposed method in running time is not that significant while dealing with small scale test selection problem. However, with the increasing of test size, for example, in complex equipment or systems, the proposed method with ICDPSO can have a significant improvement in working performance than GA.

6. Conclusions

A proper selection of the appropriate test set plays an important role in measuring the testability level, which is related to the test cost of the diagnostic for a system or equipment as well as the subsequent maintenance costs. An improved test selection optimization model based on fault ambiguity group isolation and chaotic discrete PSO is proposed in this paper, which considers the ambiguity FIR index constraint. The main contribution of this paper is the modelling and mathematical expressions of the FIR with a complicated

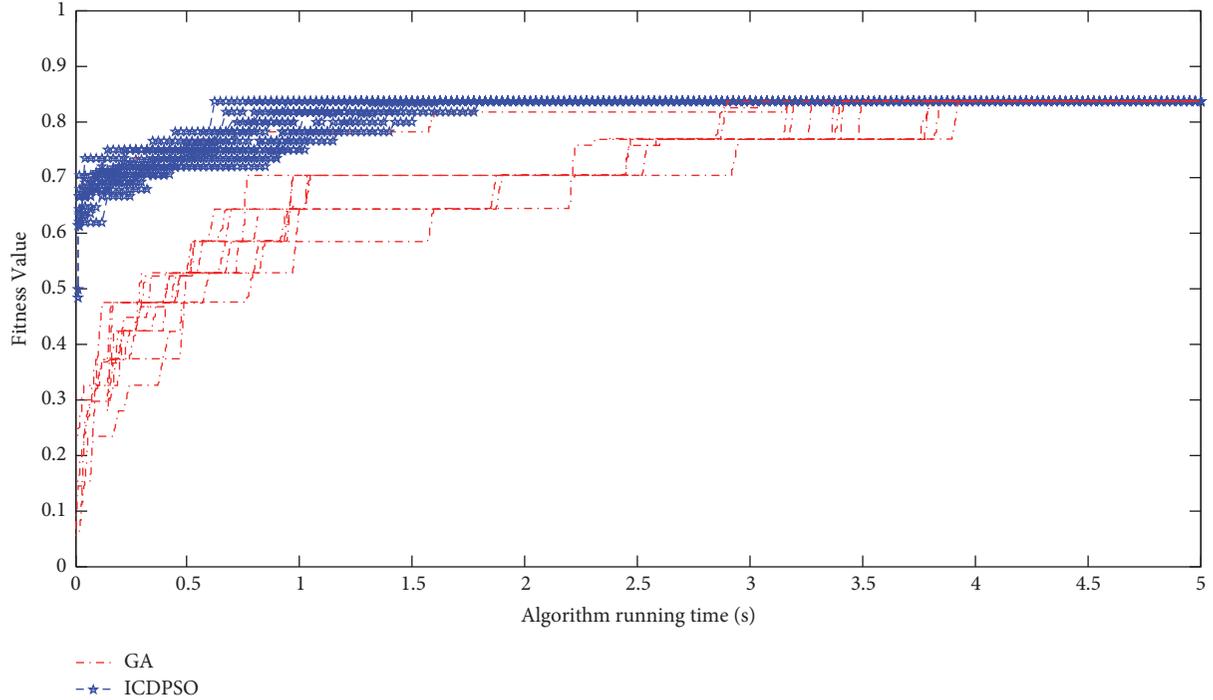


FIGURE 3: Comparison of the average convergence time of GA and ICDPSO.

TABLE 3: The experimental results of different methods.

Indicator	The method in [20]	The proposed method
FDR	96.86%	98.15%
FIR with an ambiguity degree of 1	96.57%	95.30%
FIR with an ambiguity degree of 2	100%	98.68%
FIR with an ambiguity degree of 3	N/A	100%
Test cost	6	7
Number of undetected fault modes	9	6

TABLE 4: The statistical results of GA and ICDPSO.

Algorithm	Minimum test cost	Success rate	Average running time
GA	7	100%	3.526
ICDPSO	7	100%	1.497

ambiguity degree, as well as the corresponding solving algorithm based on the chaotic discrete PSO algorithm. The FDR and FIR indexes are handled simultaneously while solving the improved test selection optimization model. The simulation result shows the feasibility and effectiveness of the proposed method, as well as the superiority in convergence.

The following work will be focused on combining the selection of test set with the designing of the diagnosis strategy, which is quite a complicated open issue related to the combination problem and the permutation problem. In addition, some new developed optimization techniques like differential evolution should be taken into consideration in the ongoing work.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The work is partially supported by the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University under Grant CX201705.

References

- [1] A. Falcoz, D. Henry, and A. Zolghadri, "Robust fault diagnosis for atmospheric reentry vehicles: a case study," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 40, no. 5, pp. 886–899, 2010.
- [2] P. Joshi, M. Imadabathuni, D. He, M. Al-Kateb, and E. Bechhofer, "Application of the Condition Based Maintenance checking system for aircrafts," *Journal of Intelligent Manufacturing*, vol. 23, no. 2, pp. 277–288, 2012.

- [3] M. E. Paté-Cornell, "Organizational aspects of engineering system safety: The case of offshore platforms," *Science*, vol. 250, no. 4985, pp. 1210–1217, 1990.
- [4] S.-M. Yang, J. Qiu, G.-J. Liu, and P. Yang, "Sensor selection and optimization for aerospace system health management under uncertainty testing," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 56, no. 4, pp. 187–196, 2013.
- [5] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, 2010.
- [6] L. Hyafil and R. L. Rivest, "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters*, vol. 5, no. 1, pp. 15–17, 1976.
- [7] R. A. Johnson, "An information theory approach to diagnosis," *IRE Transactions on Reliability and Quality Control*, vol. 9, no. 1, p. 35, 1960.
- [8] K. R. Pattipati and M. G. Alexandridis, "Application of Heuristic Search and Information Theory to Sequential Fault Diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 4, pp. 872–887, 1990.
- [9] V. Raghavan, M. Shakeri, and K. Pattipati, "Test sequencing algorithms with unreliable tests," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 29, no. 4, pp. 347–357, 1999.
- [10] C. Yang, R. Su, and B. Long, "Methods of sequential test optimization in dynamic environment," *Microelectronics Reliability*, vol. 70, pp. 112–121, 2017.
- [11] C. Yang, J. Yan, B. Long, and Z. Liu, "A novel test optimizing algorithm for sequential fault diagnosis," *Microelectronics Journal*, vol. 45, no. 6, pp. 719–727, 2014.
- [12] D. F. Jones, S. K. Mirrazavi, and M. Tamiz, "Multi-objective meta-heuristics: an overview of the current state-of-the-art," *European Journal of Operational Research*, vol. 137, no. 1, pp. 1–9, 2002.
- [13] Q. Long, C. Z. Wu, T. W. Huang, and X. Y. Wang, "A genetic algorithm for unconstrained multi-objective optimization," *Swarm and Evolutionary Computation*, 2015.
- [14] J. Ferrer, F. Chicano, and E. Alba, "Evolutionary algorithms for the multi-objective test data generation problem," *Software: Practice and Experience*, vol. 42, no. 11, pp. 1331–1362, 2012.
- [15] S. Deng, B. Jing, and H. Zhou, "Heuristic particle swarm optimization approach for test point selection with imperfect test," *Journal of Intelligent Manufacturing*, vol. 28, no. 1, pp. 37–50, 2017.
- [16] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.
- [17] R. Jiang, H. Wang, S. Tian, and B. Long, "Multidimensional fitness function DPSO algorithm for analog test point selection," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 6, pp. 1634–1641, 2010.
- [18] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 1–18, 2015.
- [19] X. Y. Chen, K. W. Chau, and A. O. Busari, "A comparative study of population-based optimization algorithms for downstream river flow forecasting by a hybrid neural network model," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 258–268, 2015.
- [20] H. Lei and K. Qin, "Optimal test selection based on improved quantum-inspired evolutionary algorithm," *Chinese Journal of Scientific Instrument*, vol. 34, no. 4, pp. 838–844, 2013.
- [21] X. Chen, J. Qiu, and G. Liu, "Optimal test selection based on hybrid BPSO and GA," *Chinese Journal of Scientific Instrument*, vol. 30, no. 8, pp. 1674–1680, 2009.
- [22] K. Balazs and L. T. Koczy, "Hierarchical-interpolative fuzzy system construction by genetic and bacterial memetic programming approaches," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 20, pp. 105–131, 2012.
- [23] K. Balazs and L. T. Koczy, "Constructing dense, sparse, and hierarchical fuzzy systems by applying evolutionary optimization techniques," *Applied And Computational Mathematics*, vol. 11, pp. 81–101, 2012.
- [24] A. Azizi, "Introducing a novel hybrid artificial intelligence algorithm to optimize network of industrial applications in modern manufacturing," *Complexity*, vol. 2017, Article ID 8728209, 18 pages, 2017.
- [25] E. I. Papageorgiou, M. F. Hatwagner, A. Buruzs, and L. T. Koczy, "A concept reduction approach for fuzzy cognitive map models in decision making and management," *Neurocomputing*, vol. 232, pp. 16–33, 2017.
- [26] X. Zhang and S. Mahadevan, "A bio-inspired approach to traffic network equilibrium assignment problem," *IEEE Transactions on Cybernetics*, 2017.
- [27] Y. Deng, S. Mahadevan, and D. Zhou, "Vulnerability assessment of physical protection systems: a bio-inspired approach," *International Journal of Unconventional Computing*, vol. 11, no. 3-4, pp. 227–243, 2015.
- [28] X. Zhang, A. Adamatzky, F. T. Chan, S. Mahadevan, and Y. Deng, "Physarum solver: a bio-inspired method for sustainable supply chain network design problem," *Annals of Operations Research*, vol. 254, no. 1-2, pp. 533–552, 2017.
- [29] H.-Y. Jung, J. H. Yoon, and S. H. Choi, "Fuzzy linear regression using rank transform method," *Fuzzy Sets and Systems*, vol. 274, pp. 97–108, 2015.
- [30] M. Namdari, J. H. Yoon, A. Abadi, S. M. Taheri, and S. H. Choi, "Fuzzy logistic regression with least absolute deviations estimators," *Soft Computing*, vol. 19, no. 4, pp. 909–917, 2015.
- [31] W.-J. Lee, H. Y. Jung, J. H. Yoon, and S. H. Choi, "The statistical inferences of fuzzy regression based on bootstrap techniques," *Soft Computing*, vol. 19, no. 4, pp. 883–890, 2015.
- [32] Z.-G. Liu, Q. Pan, J. Dezert, and A. Martin, "Adaptive imputation of missing values for incomplete pattern classification," *Pattern Recognition*, vol. 52, pp. 85–95, 2016.
- [33] R. Jirousek and P. P. Shenoy, "Compositional models in valuation-based systems," *International Journal of Approximate Reasoning*, vol. 55, no. 1, part 3, pp. 277–293, 2014.
- [34] K. Zhou, A. Martin, Q. Pan, and Z.-G. Liu, "ECMdd: Evidential c-medoids clustering with multiple prototypes," *Pattern Recognition*, vol. 60, pp. 239–257, 2016.
- [35] A. R. Yildiz, "A novel particle swarm optimization approach for product design and manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 40, no. 5-6, pp. 617–628, 2009.
- [36] A. R. Yildiz and K. N. Solanki, "Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach," *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 1-4, pp. 367–376, 2012.

- [37] O. Abedinia, A. Ghasemi, and N. Ojaroudi, "Improved time varying inertia weight PSO for solved economic load dispatch with subsidies and wind power effects," *Complexity*, vol. 21, no. 4, pp. 40–49, 2016.
- [38] B. S. Yildiz, "A comparative investigation of eight recent population-based optimisation algorithms for mechanical and structural design problems," *International Journal of Vehicle Design*, vol. 73, no. 1, pp. 208–218, 2017.
- [39] K. Balazs, Z. Horvath, and L. T. Koczy, "Different Chromosome-based Evolutionary Approaches for the Permutation Flow Shop Problem," *Acta Polytechnica Hungarica*, vol. 9, pp. 115–138, 2012.
- [40] X. Zhang, F. T. S. Chan, A. Adamatzky et al., "An intelligent physarum solver for supply chain network design under profit maximization and oligopolistic competition," *International Journal of Production Research*, vol. 55, no. 1, pp. 244–263, 2017.
- [41] Y. Tang, D. Zhou, and S. Xu, "An improved physarum solver for supply chain network with fuzzy numbers," *International Journal of Unconventional Computing*, vol. 13, no. 2, pp. 161–184, 2017.
- [42] L. T. Kóczy, P. Földesi, and B. Tüü-Szabó, "An effective Discrete Bacterial Memetic Evolutionary Algorithm for the Traveling Salesman Problem," *International Journal of Intelligent Systems*, vol. 32, no. 8, pp. 862–876, 2017.
- [43] M. Bereta, W. Pedrycz, and M. Reformat, "Analysis and design of rank-based classifiers," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3256–3265, 2013.
- [44] J. Medina, "Minimal solutions of generalized fuzzy relational equations: clarifications and corrections towards a more flexible setting," *International Journal of Approximate Reasoning*, vol. 84, pp. 33–38, 2017.
- [45] M. Bereta, P. Karczmarek, W. Pedrycz, and M. Reformat, "Local descriptors in application to the aging problem in face recognition," *Pattern Recognition*, vol. 46, no. 10, pp. 2634–2646, 2013.
- [46] P. Karczmarek, W. Pedrycz, M. Reformat, and E. Akhoundi, "A study in facial regions saliency: A fuzzy measure approach," *Soft Computing*, vol. 18, no. 2, pp. 379–391, 2014.
- [47] N. Ghadimi, "A new hybrid algorithm based on optimal fuzzy controller in multimachine power system," *Complexity*, vol. 21, no. 1, pp. 78–93, 2015.
- [48] A. R. Yildiz, "A new hybrid particle swarm optimization approach for structural design optimization in the automotive industry," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 226, no. 10, pp. 1340–1351, 2012.
- [49] A. Azadeh, N. Pourmohammad Zia, M. Saberi et al., "A trust-based performance measurement modeling using t-norm and t-conorm operators," *Applied Soft Computing*, vol. 30, pp. 491–500, 2015.
- [50] P. Julián-Iranzo, J. Medina, and M. Ojeda-Aciego, "On reductants in the framework of multi-adjoint logic programming," *Fuzzy Sets and Systems*, vol. 317, pp. 27–43, 2017.
- [51] J. H. Kim, K. Wong, G. Athanasopoulos, and S. Liu, "Beyond point forecasting: Evaluation of alternative prediction intervals for tourist arrivals," *International Journal of Forecasting*, vol. 27, no. 3, pp. 887–901, 2011.
- [52] S. Rai and K. Wong, "The use of interactive simulations to affect driving behaviour," in *Proceedings of the 3rd Annual International Conference on Computer Games Multimedia and Allied Technologies (CGAT)*, Prakash. and E., Eds., pp. 51–57, Global Science & Technology Forum, Singapore, 2010.
- [53] M. Tian, X. Gao, and C. Dai, "Differential evolution with improved individual-based parameter setting and selection strategy," *Applied Soft Computing*, vol. 56, pp. 286–297, 2017.
- [54] H.-T. Ouyang, "Multi-objective optimization of typhoon inundation forecast models with cross-site structures for a water-level gauging network by integrating ARMAX with a genetic algorithm," *Natural Hazards and Earth System Sciences*, vol. 16, no. 8, pp. 1897–1909, 2016.
- [55] K. Mahesh, P. Nallagownden, and I. Elamvazuthi, "Advanced Pareto front non-dominated sorting multi-objective particle swarm optimization for optimal placement and sizing of distributed generation," *Energies*, vol. 9, no. 12, article no. 982, 2016.
- [56] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.
- [57] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–500, 2012.
- [58] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, pp. 495–513, 2016.
- [59] J.-Y. Shi, L. Chen, and W.-W. Cui, "A new hybrid fault diagnostic method for combining dependency matrix diagnosis and fuzzy diagnosis based on an enhanced inference operator," *Circuits, Systems and Signal Processing*, vol. 35, no. 1, pp. 1–28, 2016.
- [60] Y. Cui, J. Shi, and Z. Wang, "An analytical model of electronic fault diagnosis on extension of the dependency theory," *Reliability Engineering & System Safety*, vol. 133, pp. 192–202, 2015.

Research Article

Computational Analysis of Complex Population Dynamical Model with Arbitrary Order

Fazal Haq ¹, Kamal Shah ², Ghaus ur Rahman,³
Yongjin Li ⁴ and Muhammad Shahzad ¹

¹Department of Mathematics, Hazara University Mansehra, Khyber Pakhtunkhwa, Pakistan

²Department of Mathematics, University of Malakand, Dir(L), Khyber Pakhtunkhwa, Pakistan

³Department of Mathematics and Statistics, University of Swat, Mingora, Pakistan

⁴Department of Mathematics, Sun Yat-sen University, Guangzhou, China

Correspondence should be addressed to Yongjin Li; stslj@mail.sysu.edu.cn

Received 26 September 2017; Accepted 10 December 2017; Published 3 January 2018

Academic Editor: Jesus Medina

Copyright © 2018 Fazal Haq 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 considers the approximation of solution for a fractional order biological population model. The fractional derivative is considered in the Caputo sense. By using Laplace Adomian decomposition method (LADM), we construct a base function and provide deformation equation of higher order in a simple equation. The considered scheme gives us a solution in the form of rapidly convergent infinite series. Some examples are used to show the efficiency of the method. The results show that LADM is efficient and accurate for solving such types of nonlinear problems.

1. Introduction

Nowadays in most of the research areas, the importance of fractional differential equations has been increased due to its wide range of applications in real world problems. In different scientific and engineering categories such as chemistry, mechanics, and physics applications of fractional calculus can be found. It can be also used in control theory, optimization theory, image processing, economics, and so on [1–5]. Mathematical models of fractional order are very important to study natural problems. As is known, the nature of the trajectory of the fractional order derivatives is nonlocal, describing that the fractional order derivative possesses memory effect features and any dynamical or physical system related to fractional order differential operators has a memory effect, which shows that the future states depend on the present and past states. The limitations of Caputo fractional derivative are removed while introducing Caputo-Fabrizio derivative; see [6]. This new approach was used by Singh et al. to model the dynamics of computer viruses, particularly in those cases where the physical processes do not bear plasticity, fatigue, damage, and electromagnetic hysteresis

effects. In view of the great significance of new fractional derivative, a fractional order model of chemical kinetic system with full memory effect has been studied very well. The authors examined the existence and uniqueness of the solutions for chemical kinetic system of arbitrary order by using the fixed-point theorem.

System of nonlinear differential equations are very important for mathematicians, engineers, and physicists, because in most of physical systems the input is not proportional to output in nature. In addition to the study of simple nonlinear differential equations, exact solution of the nonlinear evolution equation also plays an important role in the study of some nonlinear problems. For the exact solution there are many approaches such as Hirota's method, Darboux transformation, and Painleve expansions.

In last few years, more alternative numerical methods have been used for solving both linear and nonlinear problems of physical interest including homotopy perturbation method (HPM) [7, 8], Adomian decomposition method (ADM) [9, 10], and homotopy analysis method (HAM) [11]. Due to various applications of Lienard's equation, Kumar et al. proposed a numerical algorithm based on fractional

homotopy analysis transform method to study fractional order Lienard's equation, while recently in 2017 the authors used q -homotopy analysis method and Laplace transform approach to explore some aspects of the FitzHugh-Nagumo equation of fractional order. Laplace decomposition method was adopted in [12, 13] to deal with some other nonlinear problems while homotopy perturbation transform method is selected in [14] to explore the approximate solution. Very recently Laplace transform combined with homotopy analysis method is used to produce effective method called homotopy analysis transform method (HATM) [15, 16]. Its main purpose is to handle nonlinear physical problems. Baleanu et al. [17] have solved the fractional order optimal control problems. For the solution of system of Schrodinger-Korteweg-de Vries equation, Golmankhaneh has used the homotopy perturbation method [18]. In [19], the authors presented the following nonlinear fractional order biological population model using homotopy analysis transform method:

$$D_t^\alpha v(w, z, t) = (D_x^2 + D_z^2) v^2(w, z, t) + f(v(w, z, t)), \quad (1)$$

with initial condition as

$$v(w, z, 0) = f_0(w, z), \quad (2)$$

where $\partial/\partial t = D_t$, $\partial^2/\partial w^2 = D_w^2$, and $\partial^2/\partial z^2 = D_z^2$. Further, v represents density of the population and f denotes supply of the population due to birth and death rate. This nonlinear fractional order biological population model is obtained by replacing the first-order derivative term in the corresponding biological population model by fractional order α , where $0 < \alpha \leq 1$. The derivative is considered in Caputo's sense. The parameter involved in fractional derivative shows various responses. If $\alpha = 1$, the fractional order biological model reduces to standard biological model. In [20, 21], the authors discussed such models while exploring their different aspects. Promoting the above work, we solve model (1) by using LADM.

$$D_t^\alpha v = (D_w^2 + D_z^2) v^2 - kr v^{(a+b)} + kv^a \quad (3)$$

with the given initial condition

$$v(w, z, 0) = f_0(w, z). \quad (4)$$

Since the proposed model is highly nonlinear and contains nonlinear terms v^2 and $v^{(a+b)}$ for $a + b \neq 1$, therefore, we decompose v^2 and $v^{(a+b)}$ in terms of Adomian polynomials as

$$\begin{aligned} v(w, z, t) &= \sum_{m=0}^{\infty} v_m(w, z, t), \\ v^a(w, z, t) &= \sum_{m=0}^{\infty} T_m, \\ v^2(w, z, t) &= \sum_{m=0}^{\infty} P_m, \\ v^{a+b}(w, z, t) &= \sum_{m=0}^{\infty} Q_m, \end{aligned} \quad (5)$$

where

$$\begin{aligned} T_m &= \frac{1}{n!} \frac{d^n}{d\lambda^n} \left[\sum_{i=0}^{\infty} (\lambda^i v_i)^a \right] \Big|_{\lambda=0}, \\ P_n &= \frac{1}{n!} \frac{d^n}{d\lambda^n} \left[\sum_{i=0}^{\infty} (\lambda^i v_i)^2 \right] \Big|_{\lambda=0}, \\ Q_n &= \frac{1}{n!} \frac{d^n}{d\lambda^n} \left[\sum_{i=0}^{\infty} (\lambda^i v_i)^{a+b} \right] \Big|_{\lambda=0}. \end{aligned} \quad (6)$$

2. Preliminaries

In order to assist the readers, here in this section we recall some fundamental definitions and results from fractional calculus.

Definition 1 (see [22]). The fractional order derivative of $f(x)$ in the Caputo sense is defined as

$$\begin{aligned} {}^c D^\alpha f(x) &= I^{n-\alpha} [{}^c D^n f(x)] \\ &= \frac{1}{\Gamma(n-\alpha)} \int_0^x (x-\tau)^{n-\alpha-1} f^n(\tau) d\tau. \end{aligned} \quad (7)$$

For $n-1 \leq \alpha \leq n$, $n \in N$. In case of function $f(x, t)$, we defined Caputo derivative with respect to "x" as

$$\begin{aligned} {}^c D^\alpha f(x, t) &= I^{n-\alpha} \frac{\partial^n}{\partial x^n} f(x, t) \\ &= \frac{1}{\Gamma(n-\alpha)} \int_0^x (x-\tau)^{n-\alpha-1} \frac{\partial^n}{\partial \tau^n} f(\tau, t) d\tau. \end{aligned} \quad (8)$$

Further for $f(t) = t^\delta$, we have ${}^c D^\alpha [t^\delta] = \Gamma(\delta+1)t^{\delta-\alpha}/\Gamma(\delta+1-\alpha)$.

Definition 2 (see [22]). The Riemann-Liouville fractional order integral operator of order $\alpha > 0$ of function $f(x)$ is given as

$$I^\alpha f(x) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-\tau)^{\alpha-1} f(\tau) d\tau. \quad (9)$$

$$I^0 f(x) = f(x).$$

In case of $f(x, t)$ then the Riemann-Liouville fractional order integral operator of order $\alpha > 0$ is given by

$$I^\alpha f(x, t) = \frac{1}{\Gamma(\alpha)} \int_0^x (x-\tau)^{\alpha-1} f(\tau, t) d\tau. \quad (10)$$

$$I^0 f(x, t) = f(x, t).$$

The Riemann-Liouville fractional order integral [22] is given by

$$I^\alpha [t^\gamma] = \frac{\Gamma(\gamma+\alpha)}{\Gamma(\gamma+\alpha+1)} t^{\alpha+\gamma}, \quad (11)$$

provided that integral on the right side is point wise defined on $(0, \infty)$.

Lemma 3. Let $0 < \alpha \leq 1$ and $\Psi \in L^1(0, 1)$. Then

$$D^\alpha I^\alpha \Psi(t) = \Psi(t) \quad (12)$$

hold.

$$I^\alpha [{}^c D^\alpha \Psi(t)] = \Psi(t) - \frac{[{}^c D^{\alpha-1} \Psi(t)]_{t=0}}{\Gamma(\alpha)} t^{\alpha-1}. \quad (13)$$

holds almost everywhere on $(0, 1)$.

Definition 4 (see [19]). The Laplace transform of the Caputo derivative is given as

$$\mathcal{L}\{{}^c D^\alpha y(t)\} = s^\alpha Y(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} y^{(k)}(0), \quad (14)$$

$$n-1 < \alpha < n, \quad n \in \mathbb{N}.$$

Definition 5 (see [23]). The Mittag-Leffler function in term of power series is defined as

$$E_\alpha(t) = \sum_{k=0}^{\infty} \frac{t^k \alpha}{\Gamma(k\alpha + 1)}, \quad t \in \mathbb{C}, \quad \text{Re}(\alpha) > 0. \quad (15)$$

3. LADM for Biological Model

This section is devoted to the general procedure of the LADM for solving (3) with given initial conditions. To apply Laplace transform to model (3) we proceed as

$$\mathcal{L}\{D_t^\alpha v\} = \mathcal{L}\left\{\left(D_w^2 + D_z^2\right)v^2 - kr v^{(a+b)} + kv^a\right\}, \quad (16)$$

$$t > 0, \quad w, z \in \mathbb{R}, \quad 0 < \alpha \leq 1,$$

with the given initial condition

$$v(w, z, 0) = f_0(w, z). \quad (17)$$

From definition of Laplace transform on both sides of (3), we have

$$\begin{aligned} s^\alpha \mathcal{L}\{v(w, z, t)\} - s^{\alpha-1} v(w, z, 0) \\ = \mathcal{L}\left\{\left(D_w^2 + D_z^2\right)v^2 - kr v^{(a+b)} + kv^a\right\}. \end{aligned} \quad (18)$$

Using given initial conditions yields that

$$\begin{aligned} \mathcal{L}\{v(w, z, t)\} \\ = \frac{1}{s} f_0(w, z) \\ + \frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right)v^2 - kr v^{(a+b)} + kv^a\right\}. \end{aligned} \quad (19)$$

Therefore, after decomposing nonlinear terms in terms of Adomian polynomials and considering the unknown solutions $v = \sum_{n=0}^{\infty} v_n$, (18) can be written as

$$\begin{aligned} \mathcal{L}\left\{\sum_{n=0}^{\infty} u(x, y, t)\right\} \\ = \frac{1}{s} f_0(w, z) \\ + \frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) \sum_{n=0}^{\infty} P_n - kr \sum_{n=0}^{\infty} Q_n + k \sum_{n=0}^{\infty} T_n\right\}. \end{aligned} \quad (20)$$

Comparing terms on both sides, we get

$$\begin{aligned} \mathcal{L}\{v_0(w, z, t)\} &= \frac{1}{s} f_0(w, z), \\ \mathcal{L}\{v_1(w, z, t)\} \\ &= \frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) P_0 - kr Q_0 + kv_0\right\}, \\ \mathcal{L}\{v_2(w, z, t)\} \\ &= \frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) P_1 - kr Q_1 + kv_1\right\}, \\ \mathcal{L}\{v_3(w, z, t)\} &= \frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) P_2 - kr Q_2 + kv_2\right\} \end{aligned} \quad (21)$$

⋮

$$\begin{aligned} \mathcal{L}\{v_{n+1}(w, z, t)\} \\ = \frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) P_n - kr Q_n + kv_n\right\}, \quad n \geq 1. \end{aligned}$$

After taking Laplace inverse transform of system (21), we get

$$\begin{aligned} v_0(w, z, t) &= f_0(w, z), \\ v_1(w, z, t) \\ &= \mathcal{L}^{-1}\left[\frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) P_0 - kr Q_0 + kv_0\right\}\right], \\ v_2(w, z, t) \\ &= \mathcal{L}^{-1}\left[\frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) P_1 - kr Q_1 + kv_1\right\}\right], \\ &\vdots \\ v_{n+1}(w, z, t) \\ &= \mathcal{L}^{-1}\left[\frac{1}{s^\alpha} \mathcal{L}\left\{\left(D_w^2 + D_z^2\right) P_n - kr Q_n + kv_n\right\}\right], \end{aligned} \quad (22)$$

$$n \geq 1.$$

4. Applications

In this section we present some application of LADM to solve biological population model.

Example 6. Consider the following fractional order biological model [21]:

$$D_t^\alpha v(w, z, t) = (D_w^2 + D_z^2) v(w, z, t) - rv^2(w, z, t) + v(w, z, t), \quad (23)$$

with given initial condition

$$v(w, z, 0) = \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right). \quad (24)$$

Upon using the proposed method on (23) and comparing terms of both sides and then taking Laplace inverse transform, we get

$$\begin{aligned} v_0(w, z, t) &= \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) \\ \mathcal{L}\{v_1(w, z, t)\} &= \frac{1}{s^\alpha} \mathcal{L}\{(D_w^2 + D_z^2 - r)P_0 + v_0\} \\ v_1(w, z, t) &= \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) \frac{t^\alpha}{\Gamma(\alpha+1)}, \\ v_2(w, z, t) &= \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) \frac{t^{2\alpha}}{\Gamma(2\alpha+1)}, \\ v_3(w, z, t) &= \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) \frac{t^{3\alpha}}{\Gamma(3\alpha+1)} \\ &\vdots \end{aligned} \quad (25)$$

and so on.

The series solution is provided as

$$\begin{aligned} v &= \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) + \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) \\ &\cdot \frac{t^\alpha}{\Gamma(\alpha+1)} + \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) \frac{t^{2\alpha}}{\Gamma(2\alpha+1)} \\ &+ \dots \end{aligned} \quad (26)$$

$$\begin{aligned} v &= \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) \\ &\cdot \left[1 + \frac{t^\alpha}{\Gamma(\alpha+1)} + \frac{t^{2\alpha}}{\Gamma(\alpha+1)} + \frac{t^{3\alpha}}{\Gamma(3\alpha+1)} + \dots\right]. \end{aligned}$$

In closed form, the solution is given by

$$v(w, z, t) = \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z)\right) E_\alpha(t^\alpha), \quad (27)$$

which is the exact solution. Putting $\alpha = 1$ in (27), we get solution as

$$v(w, z, t) = \exp\left(\frac{1}{2}\sqrt{\frac{r}{2}}(w+z) + t\right). \quad (28)$$

This is the classical solution.

Example 7. Consider the following fractional order biological model [21]:

$$D_t^\alpha v(w, z, t) = (D_w^2 + D_z^2) v^2(w, z, t) + v(w, z, t), \quad (29)$$

with given initial condition

$$v(w, z, 0) = \sqrt{\sin w \sinh z}. \quad (30)$$

Applying Laplace transform on both sides, we have

$$\begin{aligned} \mathcal{L}\{D_t^\alpha v\} &= \mathcal{L}\{(D_w^2 + D_z^2) + v_n\} \\ s^\alpha \mathcal{L}\{v(w, z, t)\} - s^{\alpha-1} v(w, z, 0) &= \mathcal{L}\{(D_w^2 + D_z^2) v_n^2(w, z, t) + v_n(w, z, t)\} \\ \mathcal{L}\{v_n(w, z, t)\} &= \frac{1}{s} v(w, z, 0) \\ &+ \frac{1}{s^\alpha} \mathcal{L}\{(D_w^2 + D_z^2) v_n^2 + v_n\} \\ \mathcal{L}\{v_0(w, z, t)\} &= \frac{1}{s} \sqrt{\sin w \sinh z} \\ \mathcal{L}\{v_1(w, z, t)\} &= \frac{1}{s^\alpha} \mathcal{L}\{(D_w^2 + D_z^2) v_0^2(w, z, t) + v_0(w, z, t)\} \\ \mathcal{L}\{v_2(w, z, t)\} &= \frac{1}{s^\alpha} \mathcal{L}\{(D_w^2 + D_z^2) v_1^2(w, z, t) + v_1(w, z, t)\} \\ \mathcal{L}\{v_3(w, z, t)\} &= \frac{1}{s^\alpha} \mathcal{L}\{(D_w^2 + D_z^2) v_2^2(w, z, t) + v_2(w, z, t)\} \\ &\vdots \end{aligned} \quad (31)$$

After using Laplace inverse transform on both sides of (31), we have

$$\begin{aligned} v_0(w, z, t) &= \sqrt{\sin w \sinh z} \\ v_1(w, z, t) &= \sqrt{\sin w \sinh z} \frac{t^\alpha}{\Gamma(\alpha+1)} \\ v_2(w, z, t) &= \sqrt{\sin w \sinh z} \frac{t^{2\alpha}}{\Gamma(2\alpha+1)} \\ v_3(w, z, t) &= \sqrt{\sin w \sinh z} \frac{t^{3\alpha}}{\Gamma(3\alpha+1)} \\ &\vdots \end{aligned} \quad (32)$$

and so no. Now the series solution of problem (29) is given by

$$\begin{aligned}
v(w, z, t) &= v_0(w, z, t) + v_1(w, z, t) + v_2(w, z, t) \\
&+ \dots \\
v(w, z, t) &= \sqrt{\sin w \sinh z} + \sqrt{\sin w \sinh z} \frac{t^\alpha}{\Gamma(\alpha + 1)} \\
&+ \sqrt{\sin w \sinh z} \frac{t^{2\alpha}}{\Gamma(2\alpha + 1)} + \sqrt{\sin w \sinh z} \\
&\cdot \frac{t^{3\alpha}}{\Gamma(3\alpha + 1)} + \dots \\
v(w, z, t) &= \sqrt{\sin w \sinh z} \left[1 + \frac{t^\alpha}{\Gamma(\alpha + 1)} + \frac{t^{2\alpha}}{\Gamma(\alpha + 1)} \right. \\
&\left. + \frac{t^{3\alpha}}{\Gamma(3\alpha + 1)} + \dots \right].
\end{aligned} \tag{33}$$

The closed form is given by

$$v(w, z, t) = \sqrt{\sin w \sinh z} E_\alpha(t^\alpha). \tag{34}$$

Considering $\alpha = 1$, we get classical solution as

$$v(w, z, t) = \sqrt{\sin w \sinh z} e^t. \tag{35}$$

Example 8. Consider the following fractional order biological model.

$$D_t^\alpha v(w, z, t) = (D_w^2 + D_z^2) v(w, z, t) + kv(w, z, t), \tag{36}$$

corresponding to the initial condition

$$v(w, z, 0) = \sqrt{wz}. \tag{37}$$

Applying Laplace transform on both sides, we have

$$\begin{aligned}
&\mathcal{L} \{D_t^\alpha v(w, z, t)\} \\
&= \mathcal{L} \{(D_w^2 + D_z^2) v_n^2(w, z, t) + kv_n(w, z, t)\} \\
&s^\alpha \mathcal{L} \{v(w, z, t)\} - s^{\alpha-1} v(w, z, 0) \\
&= \mathcal{L} \{(D_w^2 + D_z^2) v_n^2(w, z, t) + v_n(w, z, t)\}
\end{aligned}$$

$$\begin{aligned}
\mathcal{L} \{v_n(w, z, t)\} &= \frac{1}{s} v(w, z, 0) \\
&+ \frac{1}{s^\alpha} \mathcal{L} \{(D_w^2 + D_z^2) v_n^2(w, z, t) + v_n(w, z, t)\} \\
\mathcal{L} \{v_0(w, z, t)\} &= \frac{1}{s} \sqrt{wz} \\
\mathcal{L} \{v_1(w, z, t)\} \\
&= k \frac{1}{s^\alpha} \mathcal{L} \{(D_w^2 + D_z^2) v_0^2(w, z, t) + v_0(w, z, t)\} \\
\mathcal{L} \{v_2(w, z, t)\} \\
&= k \frac{1}{s^\alpha} \mathcal{L} \{(D_w^2 + D_z^2) v_1^2(w, z, t) + v_1(w, z, t)\} \\
\mathcal{L} \{v_3(w, z, t)\} \\
&= k \frac{1}{s^\alpha} \mathcal{L} \{(D_w^2 + D_z^2) v_2^2(w, z, t) + v_2(w, z, t)\} \\
&\vdots
\end{aligned} \tag{38}$$

After using Laplace transform on both sides, we have

$$\begin{aligned}
v_0(w, z, t) &= \sqrt{wz} \\
v_1(w, z, t) &= k \sqrt{wz} \frac{t^\alpha}{\Gamma(\alpha + 1)} \\
v_2(w, z, t) &= k^2 \sqrt{wz} \frac{t^{2\alpha}}{\Gamma(2\alpha + 1)} \\
v_3(w, z, t) &= k^3 \sqrt{wz} \frac{t^{3\alpha}}{\Gamma(3\alpha + 1)} \\
&\vdots
\end{aligned} \tag{39}$$

and so on. The series solution of problem (36) is given by

$$\begin{aligned}
v(w, z, t) &= v_0(w, z, t) + v_1(w, z, t) + v_2(w, z, t) \\
&+ \dots \\
v(w, z, t) &= \sqrt{wz} + k \sqrt{wz} \frac{t^\alpha}{\Gamma(\alpha + 1)} + k^2 \sqrt{wz} \\
&\cdot \frac{t^{2\alpha}}{\Gamma(2\alpha + 1)} + k^3 \sqrt{wz} \frac{t^{3\alpha}}{\Gamma(3\alpha + 1)} + \dots \\
v(w, z, t) &= \sqrt{wz} \left[1 + k \frac{t^\alpha}{\Gamma(\alpha + 1)} + k^2 \frac{t^{2\alpha}}{\Gamma(\alpha + 1)} \right. \\
&\left. + k^3 \frac{t^{3\alpha}}{\Gamma(3\alpha + 1)} + \dots \right].
\end{aligned} \tag{40}$$

Hence closed form of the solution is given by

$$v(w, z, t) = \sqrt{wz} E_\alpha(kt^\alpha). \tag{41}$$

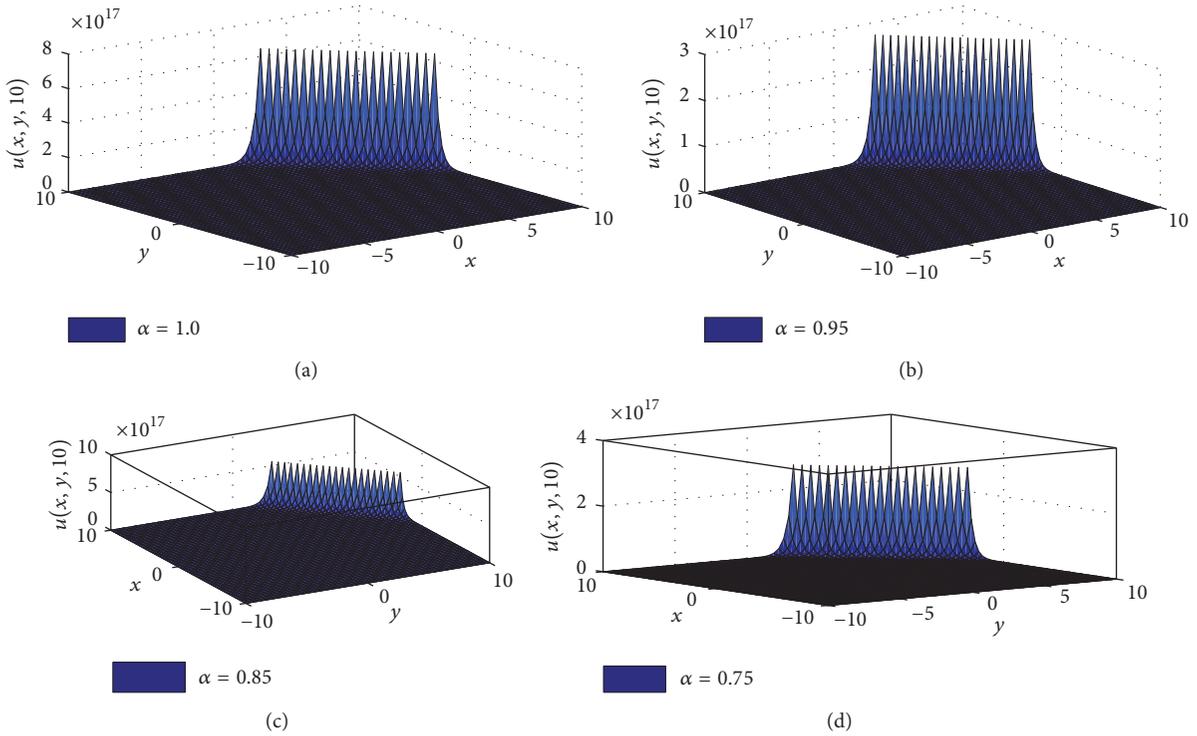


FIGURE 1: Numerical plots of approximate solutions of Example 6 at various fractional orders and using $r = 50, t = 10$.

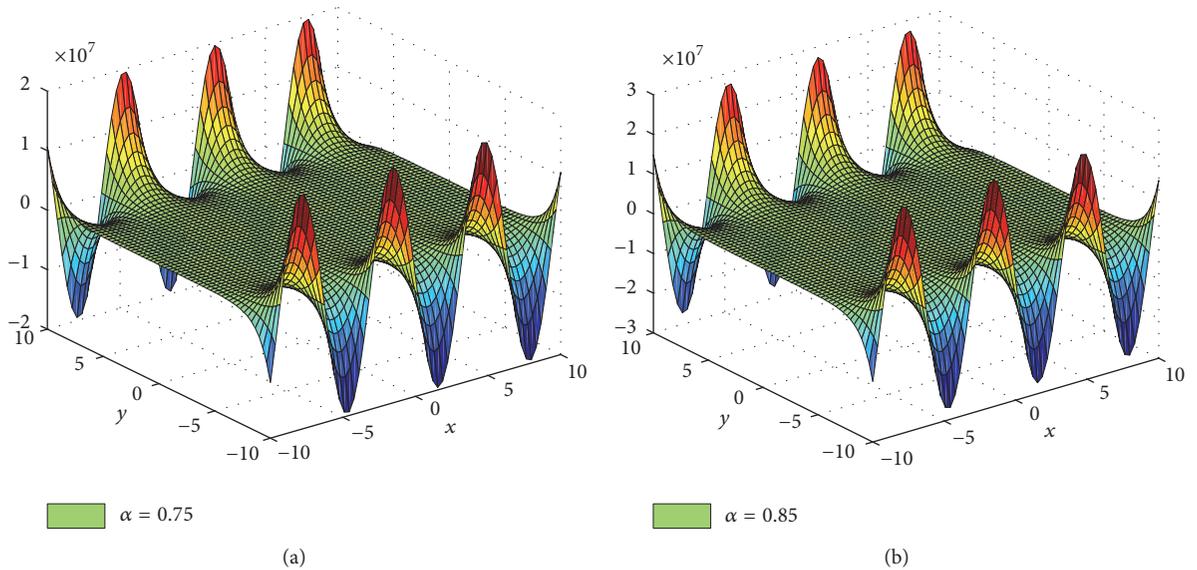


FIGURE 2: Numerical plots of approximate solutions of Example 7 at various fractional orders and using $t = 10$.

Considering $\alpha = 1$, we get classical solution as

$$v(w, z, t) = \sqrt{wz}e^t. \tag{42}$$

5. Conclusion and Discussion

In this article, the (LADM) has been applied successfully to obtain the exact solution of the generalized fractional order biological model with given initial conditions. From the

numerical plots given in Figures 1, 2, and 3 of the considered examples, we see that the procedure is efficient to obtain approximate or exact solutions to fractional order partial differential equations corresponding to various fractional order. In the current situation we have come across the exact solution for the corresponding nonlinear FPDEs.

In the 1980s George Adomian formulated a novel powerful scheme for solving nonlinear functional equations. Afterward, in the literature this method was known as Adomian

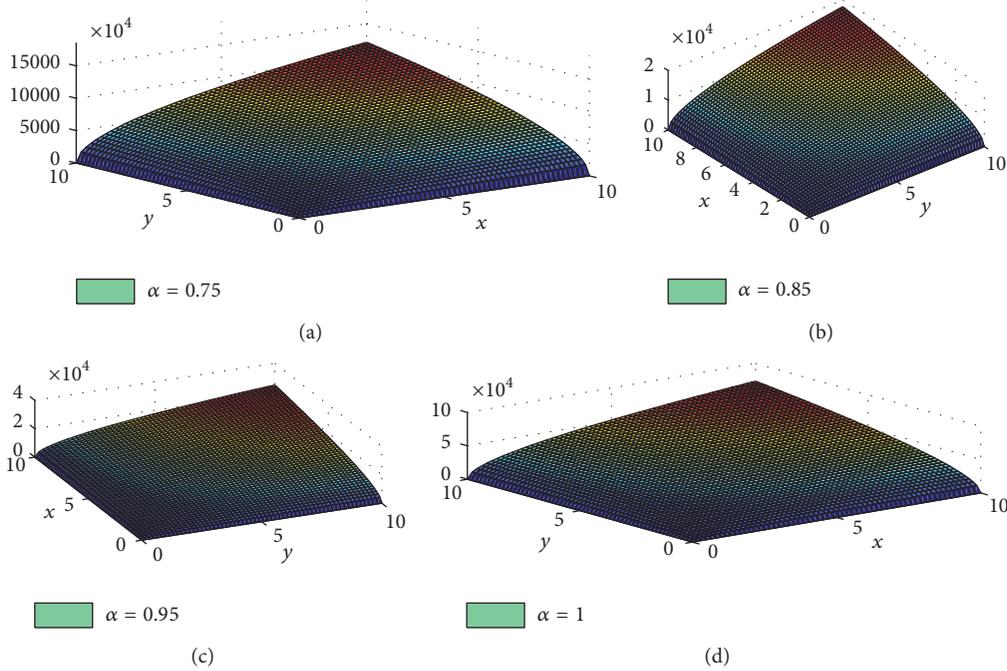


FIGURE 3: Numerical plots of approximate solutions of Example 8 at various fractional orders and using $k = 0.1, t = 10$.

decomposition method (ADM). The technique is based on the splitting of a system of differential equations solutions in the form of series of functions. Every term of the associated series is obtained from a polynomial generated by expansion of an analytic function into a power series. This is an effective tool for solution of systems of differential equations appearing in physical problems. Over the last 20 years, the Adomian decomposition approach has been applied to obtain formal solutions to a wide class of stochastic and deterministic problems involving algebraic, differential, integrodifferential, differential delay, integral, and partial differential equations. As compared to other existing numerical schemes, the main advantage of this method is that it does not require perturbation or linearization for exploring the dynamical behavior of complex dynamical systems. The LADM has done extensive work to provide analytical solution of nonlinear equations as well as solving differential equations of fractional order.

Further in this paper, the obtained results have been compared with the analytical solution and with the results obtained by Adomian decomposition method [20] and homotopy analysis method. In [20], the authors show that ADM generally does not converge, when the method is applied to highly nonlinear differential equation. Our proposed method is better than HAM, HPM, and VIM, because it needs no parameter terms to form decomposition equation, no perturbation as needed in the mention methods. Our proposed method is simple and does not require or waste extra memory like Tau-collocation method. Further from direct ADM, our method is better as we applied Laplace transform and then decompose the nonlinear terms in terms of Adomian polynomials, while in Adomian decomposition particular integral is involved, which often creates difficulties

in computation. The main advantage of this technique is that it avoids complex transformations like index reductions and leads to a simple general algorithm. Secondly, it reduces the computational work by solving only linear algebraic systems. From this method, we obtained a simple way to control the convergence region of the series solution by using a proper value of parameters. The results show that LADM is very efficient, powerful method to find the analytical solution of nonlinear differential equations.

Conflicts of Interest

There are no conflicts of interest regarding this paper.

Authors' Contributions

All authors equally contributed to this paper and approved the final version.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (11571378).

References

- [1] A. A. Kilbas, O. I. Marichev, and S. G. Samko, *Fractional Integrals and Derivatives, Theory and Applications*, Gordon and Breach, Yverdon, Switzerland, 1993.
- [2] K. S. Miller and B. Ross, *An Introduction to the Fractional Calculus and Fractional Differential Equations*, A Wiley-Interscience Publication, John Wiley & Sons, New York, NY, USA, 1993.

- [3] I. Podlubny, *Fractional Differential Equations*, vol. 198 of *Mathematics in Science and Engineering*, Academic Press, San Diego, Calif, USA, 1999.
- [4] F. Haq, K. Shah, G. Rahman, and M. Shahzad, "Existence and uniqueness of positive solution to boundary value problem of fractional," *Sindh University Research Journal*, vol. 48, no. 2, pp. 451–456, 2016.
- [5] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, *Theory and Applications of Fractional Differential Equations*, New York, NY, USA, Elsevier, 2006.
- [6] A. Atangana and D. Baleanu, "New fractional derivatives with non-local and non-singular kernel: Theory and application to heat transfer model," *Thermal Science*, vol. 20, no. 2, pp. 763–769, 2016.
- [7] D. D. Ganji, "The application of He's homotopy perturbation method to nonlinear equations arising in heat transfer," *Physics Letters A*, vol. 355, no. 4-5, pp. 337–341, 2006.
- [8] F. Haq, K. Shah, G. ur Rahman, and M. Shahzad, "Numerical solution of fractional order smoking model via laplace Adomian decomposition method," *Alexandria Engineering Journal*, 2016.
- [9] F. Haq, K. Shah, G.-U. Rahman, and M. Shahzad, "Numerical analysis of fractional order model of HIV-1 infection of CD4+ T-cells," *Computational Methods for Differential Equations*, vol. 5, no. 1, pp. 1–11, 2017.
- [10] J. Biazar, M. Gholami Porshokuhi, and B. Ghanbari, "Extracting a general iterative method from an Adomian decomposition method and comparing it to the variational iteration method," *Computers & Mathematics with Applications*, vol. 59, no. 2, pp. 622–628, 2010.
- [11] H. Kheiri, N. Alipour, and R. Dehghani, "Homotopy analysis and homotopy Pade methods for the modified Burgers-Korteweg-de Vries and the Newell-Whitehead equations," *Mathematical Sciences Quarterly Journal*, vol. 5, no. 1, pp. 33–50, 2011.
- [12] Y. Khan and N. Faraz, "A new approach to differential difference equations," *Journal of Advanced Research in Differential Equations*, vol. 2, pp. 1–12, 2010.
- [13] M. Khan and M. Hussain, "Application of Laplace decomposition method on semi-infinite domain," *Numerical Algorithms*, vol. 56, no. 2, pp. 211–218, 2011.
- [14] Y. Khan and Q. Wu, "Homotopy perturbation transform method for nonlinear equations using He's polynomials," *Computers & Mathematics with Applications*, vol. 61, no. 8, pp. 1963–1967, 2011.
- [15] M. A. Gondal, A. S. Arife, M. Khan, and I. Hussain, "An efficient numerical method for solving linear and nonlinear partial differential equations by combining homotopy analysis and transform method," *World Applied Sciences Journal*, vol. 14, no. 12, pp. 1786–1791, 2011.
- [16] M. Khan, M. A. Gondal, I. Hussain, and S. K. Vanani, "A new comparative study between homotopy analysis transform method and homotopy perturbation transform method on a semi infinite domain," *Mathematical and Computer Modelling*, vol. 55, no. 3-4, pp. 1143–1150, 2012.
- [17] D. Baleanu, O. Defterli, and O. P. Agrawal, "A central difference numerical scheme for fractional optimal control problems," *Journal of Vibration and Control*, vol. 15, no. 4, pp. 583–597, 2009.
- [18] A. K. Golmankhaneh, A. K. Golmankhaneh, and D. Baleanu, "Homotopy perturbation method for solving a system of Schrödinger-Korteweg-de Vries equations," *Romanian Reports in Physics*, vol. 63, no. 3, pp. 609–623, 2011.
- [19] D. Kumar, J. Singh, and Sushela, "Application of homotopy analysis transform method to fractional biological population model," *Romanian Reports Physics*, vol. 65, no. 1, pp. 63–75, 2013.
- [20] A. A. Arafa, S. Z. Rida, and H. Mohamed, "Homotopy analysis method for solving biological population model," *Communications in Theoretical Physics*, vol. 56, no. 5, pp. 797–800, 2011.
- [21] A. M. El-Sayed, S. Z. Rida, and A. A. Arafa, "Exact solutions of fractional-order biological population model," *Communications in Theoretical Physics*, vol. 52, no. 6, pp. 992–996, 2009.
- [22] M. Caputo, *Elasticita e Dissipazione*, Zani-Chelli, Bologna, Italy, 1969.
- [23] G. Rudolf and F. Mainardi, *Fractals and Fractional Calculus in Continuum Mechanics*, Springer, Vienna, Austria, 1997.

Research Article

Parameter Tuning for Local-Search-Based Matheuristic Methods

Guillermo Cabrera-Guerrero,¹ Carolina Lagos,¹ Carolina Castañeda,² Franklin Johnson,³ Fernando Paredes,⁴ and Enrique Cabrera⁵

¹Pontificia Universidad Católica de Valparaíso, 2362807 Valparaíso, Chile

²Instituto Tecnológico Metropolitano, Calle 73 No. 76A-374, Vía al Volador, Medellín, Colombia

³Universidad de Playa Ancha, Casilla 34-V, Valparaíso, Chile

⁴Universidad Diego Portales, 8370109 Santiago, Chile

⁵CIMFAV-Facultad de Ingeniería, Universidad de Valparaíso, 2374631 Valparaíso, Chile

Correspondence should be addressed to Guillermo Cabrera-Guerrero; guillermo.cabrera@pucv.cl

Received 20 June 2017; Revised 2 October 2017; Accepted 25 October 2017; Published 31 December 2017

Academic Editor: Kevin Wong

Copyright © 2017 Guillermo Cabrera-Guerrero 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.

Algorithms that aim to solve optimisation problems by combining heuristics and mathematical programming have attracted researchers' attention. These methods, also known as *matheuristics*, have been shown to perform especially well for large, complex optimisation problems that include both integer and continuous decision variables. One common strategy used by matheuristic methods to solve such optimisation problems is to divide the main optimisation problem into several subproblems. While heuristics are used to seek for promising subproblems, exact methods are used to solve them to optimality. In general, we say that both mixed integer (non)linear programming problems and combinatorial optimisation problems can be addressed using this strategy. Beside the number of parameters researchers need to adjust when using heuristic methods, additional parameters arise when using matheuristic methods. In this paper we focus on one particular parameter, which determines the size of the subproblem. We show how matheuristic performance varies as this parameter is modified. We considered a well-known NP-hard combinatorial optimisation problem, namely, the capacitated facility location problem for our experiments. Based on the obtained results, we discuss the effects of adjusting the size of subproblems that are generated when using matheuristics methods such as the one considered in this paper.

1. Introduction

Solving mixed integer programming (MIP) problems as well as combinatorial optimisation problems is, in general, a very difficult task. Although efficient exact methods have been developed to solve these problems to optimality, as the problem size increases exact methods fail to solve it within an acceptable computational time. As a consequence, nonexact methods such as heuristic and metaheuristic algorithms have been developed to find good quality solutions. In addition, hybrid strategies combining different nonexact algorithms are also promising ways to tackle complex optimisation problems. Unfortunately, algorithms that do not consider exact methods cannot give us any guarantee of optimality and, thus, we do not know how good (or bad) solutions found by these methods are.

One hybrid strategy that combines nonexact methods are memetic algorithms, which are population-based metaheuristics that use an evolutionary framework integrated with local search algorithms [1]. Memetic algorithms provide lifetime learning process to refine individuals in order to improve the obtained solutions every iteration or generation; their applications have been grown significantly over the years in several NP-hard optimisation problems [2]. These algorithms are part of the paradigm of memetic computation, where the concept of meme is used to automate the knowledge transfer and reuse across problems [3]. A large number of memetic algorithms can be found in the literature. Depending on its implementation, memetic algorithms might (or might not) give guarantee of local optimality: roughly speaking, if heuristic local search algorithms are considered, then no optimality guarantee is given; if exact

methods are considered as the local optimisers, then local optimality could be ensured.

To overcome the situation described above, hybrid methods that combine heuristics and exact methods to solve optimisation problems have been proposed. These methods, also known as matheuristics [4], have been shown to perform better than both heuristic and mathematical programming methods when they are applied separately. The idea of combining the power of mathematical programming with flexibility of heuristics has gained attention within researchers' community. We can find matheuristics attempting to solve problems arising in the field of logistics [5–9], health care systems [10–13], and pure mathematics [14, 15], among others. Matheuristics have been demonstrated to be very effective in solving complex optimisation problems. Some interesting surveys on matheuristics are [16, 17]. Although there is some overlapping between memetic algorithms and matheuristic ones, in this paper we have chosen to label the studied strategy as matheuristic, as we think matheuristics definition better fits the framework we are interested in.

Because of their complexity, MIP problems as well as combinatorial optimisation problems are often tackled using matheuristic methods. One common strategy to solve this class of optimisation problems is to divide the main optimisation problem into several subproblems. While heuristics are used to seek for promising subproblems, exact methods are used to solve them to optimality. One advantage of this approach is that it does not depend on the (non)linearity of the resulting subproblem. Instead, it has been pointed out that it is desirable that the resulting subproblem would be convex [11]. Having a convex subproblem would allow us to solve it to optimality, and thus comparing solutions obtained at each subproblem becomes more sensible. This strategy has been successfully applied to problems arising in fields as diverse as logistics and radiation therapy.

In this paper we aim to study the impact of parameter tuning on the performance of matheuristic methods as the one described above. To this end, the well-known capacitated facility location problem is used as an application of hard combinatorial optimisation problem. To the best of our knowledge, no paper has focused on parameter tuning for matheuristic methods.

This paper is organised as follows: Section 2 shows the general matheuristic framework we consider in this paper. Details on the algorithms that are used in this study are also shown in this section. In Section 3 the capacitated facility location problem is introduced and its mathematical model is described in Section 3.1. The experiments performed in this study are presented and the obtained results are discussed in Section 3.2. Finally, in Section 4 some conclusions are presented and the future work is outlined.

2. Matheuristic Methods

This section is twofold. We start by describing a general matheuristic framework that is used to solve both MIP problems and combinatorial optimisation problems and how it is different from other commonly used approaches such

as memetic algorithms and other evolutionary approaches. After that, we present the local-search-based algorithms we consider in this work to perform our experiments. We finish this section by introducing the parameter we will be focused on in this study.

2.1. General Framework. Equations (1a) to (1f) show the general form of MIP problems. Hereafter we will refer to this problem as the $\text{MIP}(x, y)$ problem or the *main problem*.

$$\text{MIP}(x, y) : \min_{x, y} f(x, y) \quad (1a)$$

$$\text{s.t. } g_j(x, y) \leq b_j \quad \text{for } j = 1, \dots, m, \quad (1b)$$

$$\bar{g}_j(x) \leq \bar{b}_j \quad \text{for } j = 1, \dots, \bar{m}, \quad (1c)$$

$$\bar{\bar{g}}_j(y) \leq \bar{\bar{b}}_j \quad \text{for } j = 1, \dots, \bar{\bar{m}}, \quad (1d)$$

$$y \in \{0, 1\}^o, \quad (1e)$$

$$x \in \mathbb{R}^n, \quad (1f)$$

where $f(x, y)$ is an objective function, m is the number of inequality constraints on x and y , \bar{m} is the number of inequality constraints on x , $\bar{\bar{m}}$ is the number of inequality constraints on y , o is the number of binary (≥ 0) decision variables, and n is the number of continuous decision variables. Combinatorial optimisation problems, as the one we consider in this paper, can be easily obtained by either removing the continuous decision variable from the model or making it integer (i.e., $x \in \mathbb{Z}_{\geq}^n$).

Although there exist a number of exact algorithms that can find an optimal solution for the $\text{MIP}(x, y)$, as the size of the problem increases, exact methods fail or take too long. Because of this, heuristic methods are used to obtain good quality solutions of the problem within an acceptable time. Heuristic methods cannot guarantee optimality though.

During the last two decades, the idea of combining heuristic methods and mathematical programming has received much more attention. Exploiting the advantages of each method appears to be a sensible strategy to overcome their inherent drawbacks. Several strategies have been proposed to combine heuristics and exact methods to solve optimisation problems such as the $\text{MIP}(x, y)$ problem. For instance, Chen and Ting [18] and Lagos et al. [8] combine the well-known ant colony optimisation (ACO) algorithm and Lagrangian relaxation method to solve the single source capacitated facility location problem and a distribution network design problem, respectively. In these articles, Lagrangian multipliers are updated using ACO algorithm. Another strategy to combine heuristics and exact methods is to let heuristics seek for subproblems of $\text{MIP}(x, y)$ which, in turn, are solved to optimality by some exact method. One alternative to obtain subproblems of $\text{MIP}(x, y)$ is to add a set of additional constraints on a subset of binary decision variables. These constraints are of the form $y_i = 0$, with $i \in \mathcal{I}$ and \mathcal{I} being the set of index that are restricted in subproblem $\text{MIP}_{\mathcal{I}}(x, y)$ (see (1a) to (1f)). The portion of binary decision variables y_i that are set to 0 is denoted by α (i.e., $\#\mathcal{I} = \alpha \times o$),

with o being the number of binary variables y_i . Then, the obtained subproblem, which we call $\text{MIP}_{\mathcal{F}}(x, y)$, is

$$\text{MIP}_{\mathcal{F}}(x, y) : \min_{x, y} f(x, y) \quad (2a)$$

$$\text{s.t. } g_j(x, y) \leq b_j \quad (2b)$$

for $j = 1, \dots, m$,

$$\overline{g}_j(x) \leq \overline{b}_j \quad \text{for } j = 1, \dots, \overline{m}, \quad (2c)$$

$$\overline{\overline{g}}_j(y) \leq \overline{\overline{b}}_j \quad \text{for } j = 1, \dots, \overline{\overline{m}}, \quad (2d)$$

$$y_i = 0 \quad i \in \mathcal{F}, \quad (2e)$$

$$y \in \{0, 1\}^o, \quad (2f)$$

$$x \in \mathbb{R}^n. \quad (2g)$$

In this paper we assume that a constraint on the i th resource of vector y of the form $y_i = 0$, that is, $i \in \mathcal{F}$, means that such a resource is not available for subproblem $\text{MIP}_{\mathcal{F}}(x, y)$. We can note that, as the number of constrained binary decision variables y_i associated with \mathcal{F} increases, that is, as $(1 - \alpha)$ increases, the subproblem $\text{MIP}_{\mathcal{F}}(x, y)$ becomes smaller. Similarly, as the number of constrained binary decision variables y_i associated with \mathcal{F} gets smaller, that is, $(1 - \alpha)$ decreases, the subproblem $\text{MIP}_{\mathcal{F}}(x, y)$ gets larger, as there are more available resources. It is also easy to note that, as $(1 - \alpha)$ increases, we can obtain optimal solutions of the corresponding subproblem $\text{MIP}_{\mathcal{F}}(x, y)$ relatively faster. However, quality of the obtained solutions is usually impaired as the solution space is restricted by the additional constraints associated with the set \mathcal{F} . Similarly, as $(1 - \alpha)$ gets smaller, obtaining optimal solutions of the associated subproblems might take longer but the quality of the obtained solutions is, in general, greatly improved. Finally, when $\alpha = 0$ we have that subproblem $\text{MIP}_{\mathcal{F}}(x, y)$ is identical to the main problem, $\text{MIP}(x, y)$. We assume that optimal solution of subproblem $\text{MIP}_{\mathcal{F}}(x, y)$, (\hat{x}, \hat{y}) , is also feasible for the $\text{MIP}(x, y)$ problem. Moreover, we can note that there must be a minimal set \mathcal{F} , for which an optimal solution (\hat{x}, \hat{y}) of the associated subproblem $\text{MIP}_{\mathcal{F}}(x, y)$ is also an optimal solution (x^*, y^*) of the main problem $\text{MIP}(x, y)$.

In some cases, the value of α might be predefined by the problem that is being solved. For instance, the problem of finding the best beam angle configuration for radiation delivery in cancer treatment (beam angle optimisation problem) usually sets the number of beams to be used in a beam angle configuration (see Cabrera-Guerrero et al. [11], Li et al. [19], and Li et al. [20]). This definition is made by the treatment planner and it does not take into account the algorithm performance but clinical aspects. Unlike this kind of problems, there are many other problems where the value of α is not predefined and, then, setting it to an efficient value is important for the algorithm performance. Figure 1 shows the interaction between the heuristic and the exact method.

One distinctive feature of matheuristics is that there exists an interaction between the heuristic method and the exact method. In the method depicted in Figure 1 we have that,

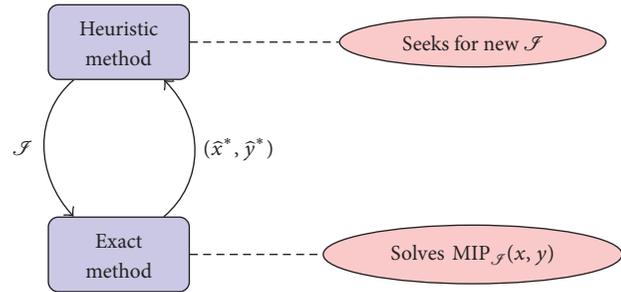


FIGURE 1: Interaction between a heuristic method and an exact method.

on one hand, the heuristic method influences the solver by passing onto it a set of constraints, \mathcal{F} , that defines the subproblem to be solved by the exact method. On the other hand, we have that the solution obtained by the solver is returned to the heuristic method such that it can be used to influence the selection of the elements in the next set of constraints; that is, the solution of a subproblem might be used by the heuristic to obtain some useful information to generate next subproblems.

As mentioned above, there is one parameter that is not part of the set of parameters of the heuristic method nor part of the set of parameters of the exact method. This parameter, which we call α , only comes up after these two methods are posed together. Then, in this paper we are interested in how the choice of α can modify the performance of the proposed strategy in terms of the quality of the obtained solutions of the main problem. Since many different matheuristic frameworks might be proposed to solve the $\text{MIP}(x, y)$ problem, we restrict this study to local-search-based matheuristics. Thus, in this study, three local-search-based matheuristic algorithms are implemented and their results compared. Additionally, we implement a very simple method which we call “blind algorithm” as a baseline for this study. Next sections explain these algorithms.

2.2. Local Search Algorithms. As mentioned in previous sections, the aim of this paper is not to provide a “state-of-the-art” algorithm to solve MIP problems but, instead, to study the effect that changing the size of subproblems has on the quality of the obtained solutions when using local-search-based matheuristic methods as the one described in Section 2.1. Thus, three local search algorithms are implemented, namely, steepest descent (SD), next descent (ND), and tabu search (TS). Local search algorithms need a neighbourhood \mathcal{N} to be defined by means of a neighbourhood movement. We define the same neighbourhood movement for the all three methods.

Since the local search algorithms move on the subproblem space, that is, the local search algorithms look for promising subproblems $\text{MIP}_{\mathcal{F}}(x, y)$, or, equivalently, look for promising sets \mathcal{F} , then we need to define a neighbourhood within the same search space. Thus, the neighbour set of \mathcal{F} , which we denote by $\mathcal{N}(\mathcal{F})$ or simply \mathcal{F}' , corresponds to all resources y_i such that they either are not considered in

```

Input:  $\alpha$  (portion of the resources included in  $\mathcal{F}$ )
Output:  $(\hat{x}, \hat{y})$  (locally optimal solution)
(1) begin
(2)    $k = 0$ ;
(3)    $\mathcal{F}_k = \text{selectBinaryVariablesRandomly}(\alpha)$ ;
(4)    $(\hat{x}, \hat{y}) = \text{solve MIP}_{\mathcal{F}_k}(x, y)$ ;
(5)   repeat
(6)     localOptimum = true;
(7)      $k = k + 1$ ;
(8)     foreach  $\mathcal{F}'_{k-1} \in \mathcal{N}(I_{k-1})$  do
(9)        $(x', y') = \text{solve MIP}_{\mathcal{F}'_{k-1}}(x, y)$ ;
(10)      If  $\text{MIP}(x', y') < \text{MIP}(\hat{x}, \hat{y})$  then
(11)         $(\hat{x}, \hat{y}) = (x', y')$ ;
(12)         $\mathcal{F}_k = \mathcal{F}'_{k-1}$ ;
(13)        localOptimum = false;
(14)   until localOptimum;
(15)   return  $(\hat{x}, \hat{y})$ ;

```

ALGORITHM 1: Steepest descent algorithm.

the optimal solution of $\text{MIP}_{\mathcal{F}}(x, y)$ or are actually part of \mathcal{F} . Since the number of resources y_i that meet these criteria might be above or below the number of elements set \mathcal{F}' must contain according to the parameter α that is being considered, we randomly select among the variables that meet these criteria until the neighbour set \mathcal{F}' is completed. This neighbourhood movement ensures that those resources y_i that are part of the optimal solution of subproblem $\text{MIP}_{\mathcal{F}}(x, y)$, that is, $y_i \notin \mathcal{F}$ and $y_i = 1$, will continue to be available for sets $\mathcal{F}' \in \mathcal{N}(\mathcal{F})$. Thus, the set of resources y_i that can potentially be part of sets $\mathcal{F}' \in \mathcal{N}(\mathcal{F})$ is defined as follows:

$$\mathcal{F}' \in \mathcal{N}(\mathcal{F}) = \{y_i : y_i \in \mathcal{F} \text{ or } (y_i \notin \mathcal{F}, \hat{y}_i = 0)\}, \quad (3)$$

where (\hat{x}, \hat{y}) is the optimal solution of $\text{MIP}_{\mathcal{F}}(x, y)$. The initial set \mathcal{F} local search methods start with is set randomly for the all three algorithms implemented in this paper.

2.2.1. Steepest Descent Algorithm. The steepest descent algorithm starts with an initial \mathcal{F}_0 for which its associated subproblem, $\text{MIP}_{\mathcal{F}_0}(x, y)$, is labelled as the current subproblem. For this current subproblem, its entire neighbourhood $\mathcal{N}(\mathcal{F}_0)$ is generated, and the neighbour that leads to the best objective function value is selected. If the best neighbour subproblem is better than the current one, then the best neighbour is set as the new current solution. If the best neighbour is not better than the current subproblem, then the algorithm stops and the optimal solution of the current subproblem is returned as a locally optimal solution of the main $\text{MIP}(x, y)$ problem. Algorithm 1 shows the pseudocode for the steepest descent algorithm implemented in this paper.

Although the steepest descent algorithm can be considered, in general, a deterministic algorithm, in the sense that, given an initial solution, it converges to the same local optima, in our implementation the algorithm does not visit all possible neighbours and, therefore, it becomes a stochastic local search. Since only one local optimum is generated at

each run, we repeat the algorithm until the time limit is reached. Same is done for both ND and TS algorithms we introduce next.

2.2.2. Next Descent Algorithm. As mentioned in Section 2.2.1, the steepest descent might take too long to converge if the size of the neighbourhood is too large or solving an individual subproblem is too time-consuming. Thus, we include in this paper a local search algorithm called next descent that aims to converge faster than the steepest descent, without a major impact on the solution quality. Algorithm 2 shows the next descent method.

Just as in the steepest descent algorithm, the next descent algorithm starts with an initial solution, which is labelled as the current solution. Then, a random element from the neighbourhood of the current solution is selected and solved, and the objective function value of its optimal solution is compared to the objective function value of the current solution. Like in the SD algorithm, if the neighbour solution is not better than the current solution, another randomly selected set \mathcal{F}' from the neighbourhood $\mathcal{N}(\mathcal{F}_{k-1})$ is generated and compared to the current solution. Unlike the SD algorithm, in the ND algorithm, if the neighbour is better than the current solution, then the neighbour is labelled as the new current solution and no other $\mathcal{F}' \in \mathcal{N}(\mathcal{F}_{k-1})$ is visited. The algorithm repeats these steps until the entire neighbourhood has been computed with no neighbour resulting in a better solution than the current one, in which case the algorithm stops.

2.2.3. Tabu Search Algorithm. Unlike the algorithms described in the previous sections, tabu search is a local search technique guided by the use of adaptive or flexible memory structures [5], and thus, it is inherently a stochastic local search algorithm. The variety of the tools and search principles introduced and described in [21] are such that the TS can be considered as the seed of a general framework for modern heuristic search [22]. We include TS as it has

```

Input:  $\alpha$  (portion of the resources included in  $\mathcal{J}$ )
Output:  $(\hat{x}, \hat{y})$  (locally optimal solution)
(1) begin
(2)  $k = 0$ ;
(3)  $\mathcal{J}_k = \text{selectBinaryVariablesRandomly}(\alpha)$ ;
(4)  $(\hat{x}, \hat{y}) = \text{solve MIP}_{\mathcal{J}_k}(x, y)$ ;
(5) repeat
(6)   localOptimum = true;
(7)    $k = k + 1$ ;
(8)   foreach  $\mathcal{J}'_{k-1} \in \mathcal{N}(I_{k-1})$  do
(9)      $(x', y') = \text{solve MIP}_{\mathcal{J}'_{k-1}}(x, y)$ ;
(10)    if  $\text{MIP}(x', y') < \text{MIP}(\hat{x}, \hat{y})$  then
(11)       $(\hat{x}, \hat{y}) = (x', y')$ ;
(12)       $\mathcal{J}_k = \mathcal{J}'_{k-1}$ ;
(13)      localOptimum = false;
(14)    break;
(15) until localOptimum;
(16) return  $(\hat{x}, \hat{y})$ ;

```

ALGORITHM 2: Next descent algorithm.

been applied to several combinatorial optimisation problems (see, e.g., [5, 23–27]) including, of course, mixed integer programming problems as the one we consider in this study.

As in the algorithms introduced above, TS also starts with an initial set of constraints \mathcal{J}_0 for which its associated subproblem $\text{MIP}_{\mathcal{J}_0}(x, y)$ is solved. Then, as in the SD algorithm, the “entire” neighbourhood of the current solution is computed. As explained before, since the neighbourhood \mathcal{N} has a stochastic component, it is actually not possible to generate the entire neighbourhood of a solution; however, we use the term “entire” to stress the fact that all the generated neighbours of \mathcal{N} are solved. This is different from the ND algorithm, where not all the generated neighbours are necessarily solved. The number of generated neighbours, n_s , is, as in the algorithms above, set equal to 10. Moreover, TS implements a list called *tabu list* that aims to avoid cycles during the search. Each time a neighbourhood is generated, the warehouses removed from $I_{(k-1)}$ are marked as tabu. Once we have solved the subproblems $\text{MIP}_{\mathcal{J}'_k}(x, y)$, with $\mathcal{J}'_k \in \mathcal{N}(I_k)$, its optimal solutions are ranked and the one with the best objective function value is chosen as candidate solution for the next iteration. If the candidate solution was generated using a movement within the tabu list, it should be discarded, and the next best neighbour of the list should be chosen as the new candidate solution. However, there is one exception to this rule: if the candidate solution is within the tabu list but its objective function value is better than the best objective function found so far by the algorithm, then the so called *aspiration criterion* is invoked and the candidate solution is set as the new current solution and passed on to the next iteration.

Unlike the algorithms introduced above, TS does not require next current solution to be better than the previous one. This means that it is able to avoid local optimal by choosing solutions that are more expensive as they allows the algorithm to visit other (hopefully promising) areas in the search space. However, in case the algorithm cannot make

any improvement after a predefined number of iterations, a diversification mechanism is used to get out from low-quality neighbourhoods and “jump” to other neighbourhoods. The diversification mechanism implemented here is a restart method, which set the current solution to a randomly generated solution without losing the best solution found so far. Termination criterion implemented here is the time limit.

The tabu search implemented in this paper is as follows.

As Algorithm 3 shows, tabu search requires the following parameters:

- (i) *Time limit*: total time the algorithm will perform.
- (ii) *DiversBound*: total number of iteration without improvements on the best solution before diversification criterion (restart method) is applied.
- (iii) *TabuListSize* (ts): size of *tabuList*. Number of iterations for which a specific movement remains banned.

2.2.4. Blind Algorithm. We finally implement a very simple heuristic method we call *blind* as it moves randomly at each iteration. Pseudocode of this method is presented in Algorithm 4.

As we can see, no additional intelligence is added to the blind algorithm. It is just a random search that, after a predefined number of iterations (or any other “stop criterion”), returns the best solution it found during its search. Thus, we can consider this algorithm as a baseline of this study.

We apply the all four algorithms described in this section to two prostate cases. Details on this case and the obtained results are presented in the next section.

2.3. Subproblem Sizing. All the algorithms above perform very different as the value of the input parameter α varies. On the one hand, setting α to a very small value provokes that either the solver fails to solve the subproblem because of lack of memory or it takes too long to find an optimal

```

Input:  $\alpha$  (portion of the resources included in  $\mathcal{J}^k$ )
Output:  $(\hat{x}, \hat{y})$  (approximately optimal solution)
(1) begin
(2)  $k = 0$ ;
(3)  $\mathcal{J}_k = \text{selectBinaryVariablesRandomly}(\alpha)$ ;
(4)  $(x^c, y^c) = \text{solve MIP}_k(x, y)$ ; // set current sol
(5)  $(\hat{x}, \hat{y}) = (x^c, y^c)$ ; // set best sol
(6)  $\mathcal{Y} = \emptyset$ ; // tabu list initially empty
(7)  $\text{noImprovementCounter} = 0$ ;
(8) repeat
(9)    $\mathcal{C} = \emptyset$ ;
(10)   $k = k + 1$ ;
(11)  foreach  $\mathcal{J}'_{k-1} \in \mathcal{N}(I'_{k-1})$  do
(12)     $(x^n, y^n) = \text{solve MIP}_{\mathcal{J}'_{k-1}}(x, y)$ ;
(13)     $\mathcal{C} = \mathcal{C} \cup \{(x^n, y^n)\}$ ;
(14)  Sort( $\mathcal{C}$ );
(15)  foreach  $\{(x^n, y^n)\} \in \mathcal{C}$  do
(16)    if  $\text{isTabu}(x^n, y^n)$  then
(17)      if  $\text{MIP}(x^n, y^n) < \text{MIP}(\hat{x}, \hat{y})$  then
(18)         $\mathcal{Y} = \text{updateTabuList}(\mathcal{Y}, (x^c, y^c), (x^n, y^n), \text{ts})$ ;
(19)         $(x^c, y^c) = (x^n, y^n)$ ;
(20)         $(\hat{x}, \hat{y}) = (x^n, y^n)$ ;
(21)         $\mathcal{J}_k = \mathcal{J}'_{k-1}$ ;
(22)         $\text{noImprovementCounter} = 0$ ;
(23)        break;
(24)    else
(25)       $\mathcal{Y} = \text{updateTabuList}(\mathcal{Y}, (x^c, y^c), (x^n, y^n))$ ;
(26)       $(x^c, y^c) = (x^n, y^n)$ ;
(27)       $\mathcal{J}_k = \mathcal{J}'_{k-1}$  if  $\text{MIP}(x^n, y^n) < \text{MIP}(\hat{x}, \hat{y})$  then
(28)         $(\hat{x}, \hat{y}) = (x^n, y^n)$ ;
(29)         $\text{noImprovementCounter} = 0$ ;
(30)    else
(31)       $\text{noImprovementCounter} ++$ ;
(32)    break;
(33)  if  $\text{noImprovementCounter} \geq \text{diversBound}$  then
(34)     $\mathcal{J}_k = \text{selectBinaryVariablesRandomly}(\alpha)$ ;
(35)  until  $\text{time limit is reached}$ ;
(36)  return  $(\hat{x}, \hat{y})$ ;

```

ALGORITHM 3: Tabu search algorithm.

```

// To find an (approximately) optimal solution for  $\text{MIP}(x, y)$ 
Input:  $\alpha$  (portion of the resources included in  $\mathcal{J}^k$ )
Output:  $(\hat{x}, \hat{y})$  (approximately optimal solution)
(1) begin
(2)  $k = 0$ ;
(3)  $\mathcal{J}_k = \text{selectBinaryVariablesRandomly}(\alpha)$ ;
(4)  $(\hat{x}, \hat{y}) = \text{solve}(\text{MIP}_{\mathcal{J}_k}(x, y))$ ;
(5) while  $\text{!stopCriterion}$  do
(6)    $k = k + 1$ ;
(7)    $\mathcal{J}_k = \text{selectBinaryVariablesRandomly}(\alpha)$ ;
(8)    $(x^c, y^c) = \text{solve}(\text{MIP}_{\mathcal{J}_k}(x, y))$ ;
(9)   if  $\text{MIP}(x^c, y^c) < \text{MIP}(\hat{x}, \hat{y})$  then
(10)     $(\hat{x}, \hat{y}) = (x^c, y^c)$ ;
(11)     $\mathcal{J}_k = \mathcal{J}_{k-1}$ 
(12)  return  $(\hat{x}, \hat{y})$ ;

```

ALGORITHM 4: Matheuristic method using the blind algorithm.

solution of the subproblem. On the other hand, setting α close to the total number of binary decision variables provokes that the algorithm fails to find a solution for the generated subproblems as there is no feasible solution to it (i.e., the subproblems are too restrictive). Thus, we have to find a value of α such that exact methods can solve the obtained subproblems within an acceptable time. Further, α should allow local search methods to iterate as much as needed. Solving the obtained subproblems within few seconds is critical to matheuristic methods as they usually need several iterations before to converge to a good quality solution. This is especially true for matheuristics that consider local search or population-based heuristic methods. Then, there is a trade-off between the quality of the solution of subproblems and the time that is needed to generate such solutions. Therefore, finding a value of α that gives us a good compromise between these two aspects is critical for the overall performance of the matheuristic methods explained before. It is interesting that the problem of finding *efficient* values of α might be seen as a multiobjective optimisation problem.

In next section we explain the experiments that we perform to study how the choice of α hits local-search-based matheuristics performance. Based on the results, we draft some guidelines to set value of parameter α at the end of next section.

3. Computational Experiments

This section starts briefly introducing the problem we consider in this paper. Then, the experiments performed here are presented and their results are discussed.

3.1. The Capacitated Facility Location Problem. The capacitated facility location problem (CFLP) is a well-known problem in combinatorial optimisation. The CFLP has been shown to be NP-hard [28]. The problem consists of selecting specific sites at which to install plants, warehouses, and distribution centres while assigning customers to service facilities and interconnecting facilities using flow assignment decisions [23]. In this study we consider the CFLP problem to evaluate the performance of a very simple matheuristic algorithm. We consider a two-level supply chain in which a single plant serves a set of warehouses, which in turn serve a set of end customers or retailers. Figure 2 shows the basic configuration of our supply chain. Thus, the goal is to find a set of locations that serves the entire set of customers in an optimal way. As Figure 2 shows, each customer (or cluster) is served only by one warehouse.

The optimisation model considers the *installation* cost (i.e., the cost associated with opening a specific warehouse) and transportation or *allocation* cost (i.e., the cost of transporting one item from a warehouse to a customer). The mathematical model for the CFLP is

$$\text{CFLP}(x, y) : \min_{x, y} \sum_{i=1}^n (f_i y_i) + \sum_{i=1}^n \sum_{j=1}^m (c_{ij} d_j x_{ij}) \quad (4a)$$

$$\text{s.t.} \quad \sum_{j=1}^m d_j x_{ij} \leq I_i^{\text{cap}} y_i \quad \forall i = 1, \dots, n, \quad (4b)$$

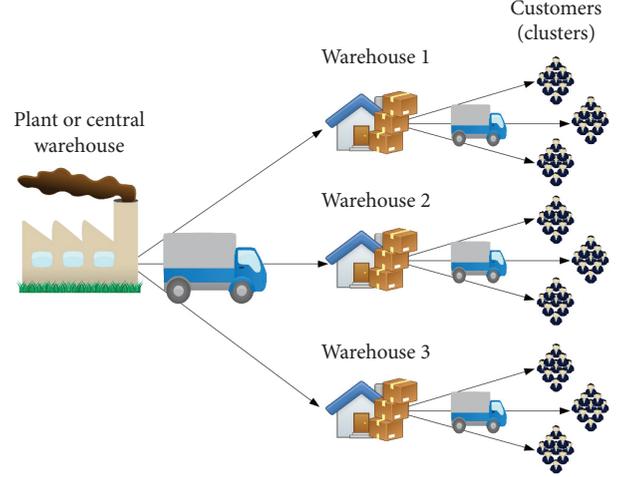


FIGURE 2: Distribution network structure considered in this study. It consists of one central plant, a set of potential warehouses, and a set of customers or retailers [23].

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j = 1, \dots, m, \quad (4c)$$

$$x_{ij} \leq y_i \quad (4d)$$

$$\forall i = 1, \dots, n; \forall j = 1, \dots, m,$$

$$y_i, x_{ij} \in \{0, 1\} \quad (4e)$$

$$\forall i = 1, \dots, n; \forall j = 1, \dots, m.$$

Equation (4a) is the total system cost. The first term is the fixed setup and operating cost when opening warehouses. The second term is the daily transport cost between warehouse and customers which depends on the customer demand d and distance c_{ij} between warehouse i and customer j . Inequality (4b) ensures that total demand of warehouse i will never be greater than its capacity I_i^{cap} . Equation (4c) ensures that customers are served by only one warehouse. Equation (4d) makes sure that customers are only allocated to available warehouses. Finally, (4e) states integrality (0–1) for the binary variables x_{ij} and y_i . Other versions of the CFLP relax this constraints by making $x \in \mathbb{R}^{n \times m}$ with $0 \leq x_{ij} \leq 1$. In that case, the CFLP would be just as the MIP(x, y) problem.

3.2. Experiments. In this paper three benchmarks for the CFLP are considered. The first benchmark corresponds to problem sets \mathcal{A} , \mathcal{B} , and \mathcal{C} from the OR Library [29]. Instances in problem sets \mathcal{A} , \mathcal{B} , and \mathcal{C} consider 1,000 clients and 100 warehouses. Warehouses capacity for instances \mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_3 , and \mathcal{A}_4 are equal to 8,000; 10,000; 12,000; and 14,000, respectively. Warehouses capacity for instances \mathcal{B}_1 , \mathcal{B}_2 , \mathcal{B}_3 , and \mathcal{B}_4 are equal to 5,000; 6,000; 7,000; and 8,000, respectively. Warehouses capacity for instances \mathcal{C}_1 , \mathcal{C}_2 , \mathcal{C}_3 , and \mathcal{C}_4 are equal to 5,000; 5,750; 6,500; and 7,250, respectively. Table 1 shows these instances and their corresponding optimal values obtained by the MILP solver. Column t shows

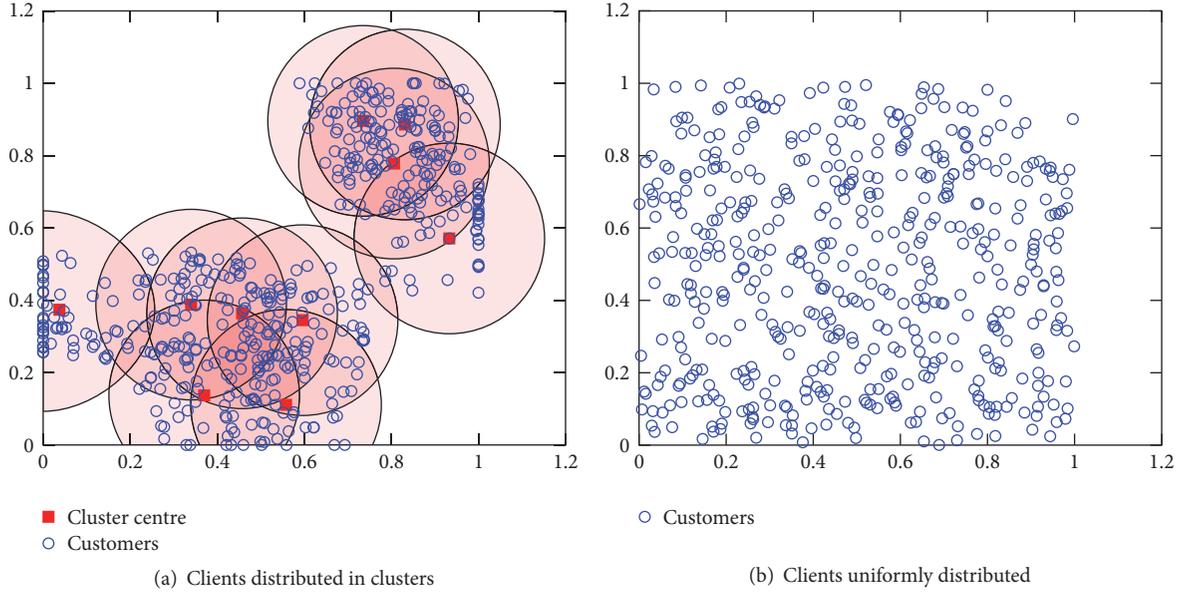


FIGURE 3: Example of the instances considered in this study.

TABLE 1: Instances for the first benchmark used in this study (OR Library).

Instances	$ I \times J $	OR Library		
		I^{cap}	$f(x^*)$	t (sec)
\mathcal{A}_1	1,000 × 100	8,000	19,240,822.45	1,745
\mathcal{A}_2	1,000 × 100	10,000	18,438,046.54	1,778
\mathcal{A}_3	1,000 × 100	12,000	17,765,201.95	199
\mathcal{A}_4	1,000 × 100	14,000	17,160,439.01	8
\mathcal{B}_1	1,000 × 100	5,000	13,656,379.58	136
\mathcal{B}_2	1,000 × 100	6,000	13,361,927.45	151
\mathcal{B}_3	1,000 × 100	7,000	13,198,556.43	256
\mathcal{B}_4	1,000 × 100	8,000	13,082,516.50	27
\mathcal{C}_1	1,000 × 100	5,000	11,646,596.97	155
\mathcal{C}_2	1,000 × 100	5,750	11,570,340.29	70
\mathcal{C}_3	1,000 × 100	6,500	11,518,743.74	31
\mathcal{C}_4	1,000 × 100	7,250	11,505,767.39	24

the time needed by the MILP solver to reach the optimal solution.

The second benchmark is a set of instances where clients and warehouses are uniformly distributed over an imaginary square of 100×100 [distance units]² (see Figure 3(a)). We call this set DS1_U . The number of clients considered in instances belonging to set DS1_U ranges from 500 to 1000 (500, 600, 700, 800, 900, and 1,000) while the number of warehouses considered varies between 500 and 1,000. Thus, set DS1_U consists of 12 problem classes ($500 \times 500, 500 \times 600, \dots, 500 \times 900, 500 \times 1000, 1000 \times 500, \dots, 1000 \times 1000$). For each problem class, 10 instances are randomly generated using the procedure proposed in [30] and that was also used in [5, 23]. We do this in order to minimise any instance dependant effect. Table 2 shows the average values for each class of problems.

Finally, a third benchmark consisting on clients that are organised in clusters is considered. We call this benchmark

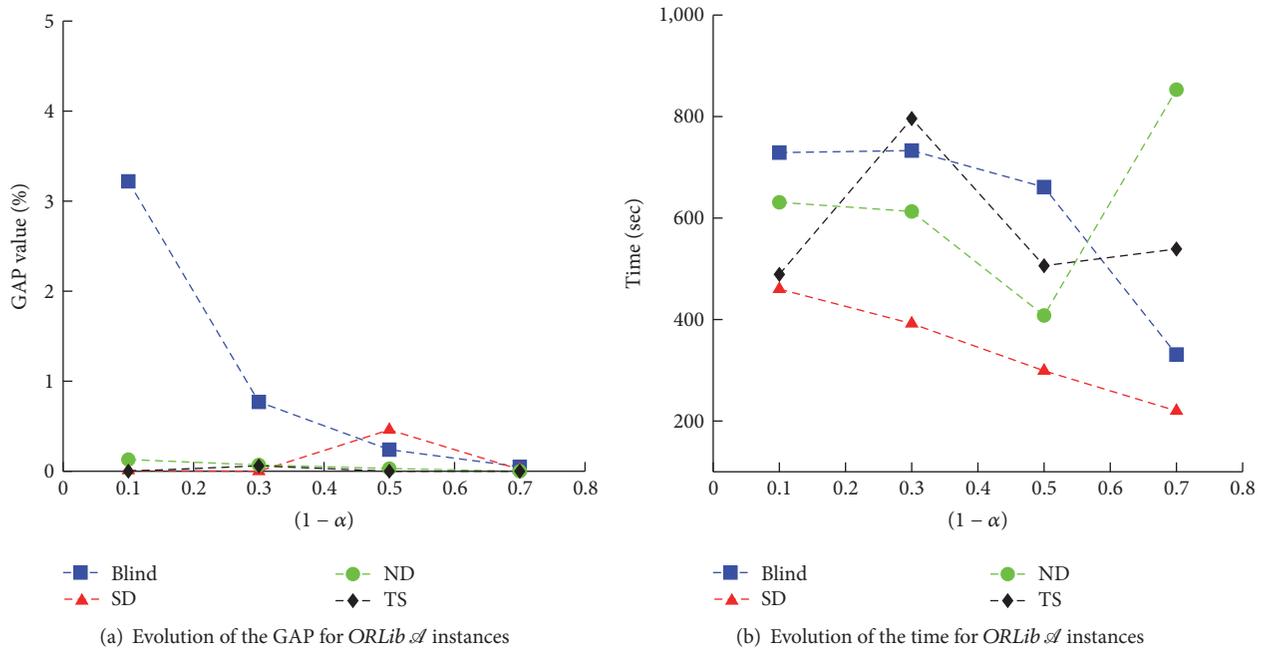
DS1_C (see Figure 3(b)). Instances in DS1_C are generated very similar to the ones in DS1_U . The only difference is that clients in DS1_C are not uniformly distributed and, thus, those warehouses that are within (or very close to) a cluster have an installation cost slightly higher than those warehouses that are far away from the clusters.

Table 2 shows the results obtained by the MIP solver from Gurobi when solving each instance of DS1_C . As we can see, the MIP solver is able to solve all the instances to optimality. Further, the solver finds the optimal solution for almost all the instances in less than 1,800 secs. Columns t_{avg} , t_{min} , and t_{max} in Table 2 show the average time and both minimum and maximum times, respectively. This is because we solve 10 different instances for each instance class. As mentioned before, we do this to avoid any instance dependent effect.

After we have solved the problem using the MIP solver, we apply the all three local-search-based matheuristics and

TABLE 2: Instances for the $DS1_U$ and $DS1_C$ benchmarks.

Instances	$ I \times J $	$f(x^*)$	$DS1_U$			$DS1_C$			
			t_{avg}	t_{min}	t_{max}	$f(x^*)$	t_{avg}	t_{min}	t_{max}
(1)	500×500	120670.3	170	95	287	123900.7	191	99	410
(2)	500×1000	110534.5	267	194	356	113527.1	262	201	410
(3)	600×500	149539.9	261	105	424	149343.2	287	120	522
(4)	600×1000	141847.7	399	309	568	137451.4	331	255	501
(5)	700×500	182434.9	427	128	674	181149.3	292	96	645
(6)	700×1000	161882.7	726	263	1515	161872.9	596	306	1673
(7)	800×500	209319.5	455	227	748	207036.9	343	131	648
(8)	800×1000	193647.7	761	399	1679	188859.7	1040	590	1828
(9)	900×500	243403.7	391	179	681	236287.7	428	178	1024
(10)	900×1000	211302.4	1228	672	1787	215410.7	1158	507	1564
(11)	1000×500	270449.9	448	189	1484	265805.9	407	154	642
(12)	1000×1000	243625.0	1187	670	1791	239519.8	1419	588	2653

FIGURE 4: Average results obtained by the local search algorithms for each value of $(1 - \alpha)$ for $ORLib \mathcal{A}$ instances.

the blind algorithm to each instance and allow them to run for 2,000 secs. The proposed algorithms solve each instance 10 times for each value of $(1 - \alpha)$. Table 3 shows the results obtained by the local-search-based matheuristic methods for instances \mathcal{A} , \mathcal{B} , and \mathcal{C} for four different values of $(1 - \alpha)$. As expected, the blind algorithm consistently obtains GAP values much higher than the other three methods. Further, local-search-based algorithms are able to find the optimal solution for the majority of the instances. While the average GAP for the blind algorithm is 1.65%, the average GAP for the SD, ND, and TS methods is 0.33%, 0.36%, and 0.15%, respectively. As expected, the best average value is obtained by the TS algorithm while difference between SD and ND algorithms is negligible. Regarding the time needed by the algorithms to converge, the blind algorithm

is the one that takes longer (770 secs). As mentioned before, the ND algorithm is, in average, the fastest one with 299 seconds, while the SD algorithm almost doubles this time with 547 seconds, in average. TS algorithm takes 371 seconds, in average, before convergence.

Figures 4(a), 5(a), and 6(a) show the evolution of the GAP, as parameter $(1 - \alpha)$ increases. As we expected, the larger the value of $(1 - \alpha)$ is set to, the smaller the GAP. Also, it is clear that the blind algorithms obtain higher GAP values than the other three algorithms considered in this study. As we mentioned before, the blind algorithm works as our baseline algorithm. It is interesting to note that, for all algorithms, the worst performance is obtained for $(1 - \alpha) = 0.1$. This is mainly because for this value not enough facilities are available and the algorithms open a set

TABLE 3: Results obtained by the local search methods for the OR Library's instances.

Inst	$(1 - \alpha)$	Blind			SD			ND			TS		
		GAP	Time	Time	GAP	Time	Time	GAP	Time	Time	GAP	Time	Time
\mathcal{A}_1	0.1	5,22%	622	248	9,65%	248	9,45%	1424	6,73%	785			
	0.3	1,63%	1124	1240	0,00%	1240	0,00%	489	0,00%	736			
	0.5	0,32%	664	1095	0,00%	1095	0,00%	274	0,24%	700			
	0.7	0,10%	745	366	0,00%	366	0,00%	138	0,00%	1476			
\mathcal{A}_2	0.1	4,31%	463	833	0,00%	833	0,00%	1415	0,00%	866			
	0.3	1,22%	893	366	0,00%	366	0,19%	1290	0,19%	615			
	0.5	0,52%	987	52	1,83%	52	0,09%	956	0,00%	745			
	0.7	0,11%	282	378	0,09%	378	0,00%	1562	0,00%	743			
\mathcal{A}_3	0.1	1,81%	841	341	0,00%	341	0,31%	65	0,00%	686			
	0.3	0,23%	503	602	0,00%	602	0,00%	693	0,00%	1083			
	0.5	0,12%	880	431	0,00%	431	0,00%	442	0,00%	659			
	0.7	0,00%	272	151	0,00%	151	0,00%	211	0,00%	211			
\mathcal{A}_4	0.1	1,56%	990	25	0,00%	25	0,00%	84	0,00%	84			
	0.3	0,00%	411	8	0,00%	8	0,00%	31	0,00%	31			
	0.5	0,00%	111	2	0,00%	2	0,00%	14	0,00%	14			
	0.7	0,00%	23	1	0,00%	1	0,00%	7	0,00%	7			
\mathcal{B}_1	0.1	5,22%	622	639	0,03%	639	0,23%	960	0,01%	319			
	0.3	1,63%	1124	588	0,01%	588	0,08%	436	0,03%	1452			
	0.5	0,32%	664	707	0,00%	707	0,03%	216	0,00%	605			
	0.7	0,10%	745	347	0,01%	347	0,00%	1630	0,00%	1192			
\mathcal{B}_2	0.1	4,31%	463	806	1,44%	806	2,75%	222	0,09%	714			
	0.3	1,22%	893	135	0,00%	135	0,00%	69	0,00%	1165			
	0.5	0,52%	987	186	0,00%	186	0,00%	196	0,05%	530			
	0.7	0,11%	282	112	0,00%	112	0,05%	315	0,00%	261			
\mathcal{B}_3	0.1	1,81%	841	403	0,40%	403	0,43%	1270	0,30%	142			
	0.3	0,23%	503	96	0,00%	96	0,00%	181	0,00%	267			
	0.5	0,12%	880	588	0,00%	588	0,00%	236	0,00%	564			
	0.7	0,00%	272	179	0,00%	179	0,00%	316	0,00%	316			
\mathcal{B}_4	0.1	1,56%	990	1309	0,00%	1309	0,32%	144	0,32%	144			
	0.3	0,00%	411	24	0,00%	24	0,00%	360	0,00%	360			
	0.5	0,00%	111	57	0,00%	57	0,00%	381	0,00%	381			
	0.7	0,00%	23	28	0,00%	28	0,00%	349	0,00%	349			
\mathcal{C}_1	0.1	8,09%	712	1220	9,70%	1220	8,14%	92	10,27%	133			
	0.3	3,20%	919	657	0,00%	657	0,00%	486	0,00%	394			
	0.5	1,99%	1105	173	0,00%	173	0,00%	496	0,00%	1469			
	0.7	0,48%	863	239	0,00%	239	0,00%	355	0,00%	247			

TABLE 3: Continued.

Inst	$(1 - \alpha)$	Blind		SD		ND		TS	
		GAP	Time	GAP	Time	GAP	Time	GAP	Time
\mathcal{E}_2	0.1	5,74%	937	1,42%	1254	2,32%	1397	0,70%	84
	0.3	1,64%	934	0,00%	136	0,00%	107	0,00%	172
	0.5	0,35%	562	0,00%	1225	0,00%	248	0,00%	537
	0.7	0,02%	766	0,00%	1294	0,00%	164	0,00%	169
\mathcal{E}_3	0.1	5,17%	979	1,11%	1254	1,54%	819	0,20%	585
	0.3	1,04%	993	0,00%	73	0,00%	38	0,00%	272
	0.5	0,22%	906	0,00%	587	0,00%	64	0,00%	1387
	0.7	0,04%	740	0,00%	551	0,00%	149	0,00%	779
\mathcal{E}_4	0.1	4,86%	663	1,35%	153	0,34%	523	0,85%	313
	0.3	0,65%	711	0,03%	7	0,03%	35	0,03%	107
	0.5	0,08%	821	0,00%	18	0,03%	11	0,03%	13
	0.7	0,02%	228	0,03%	8	0,03%	22	0,00%	29

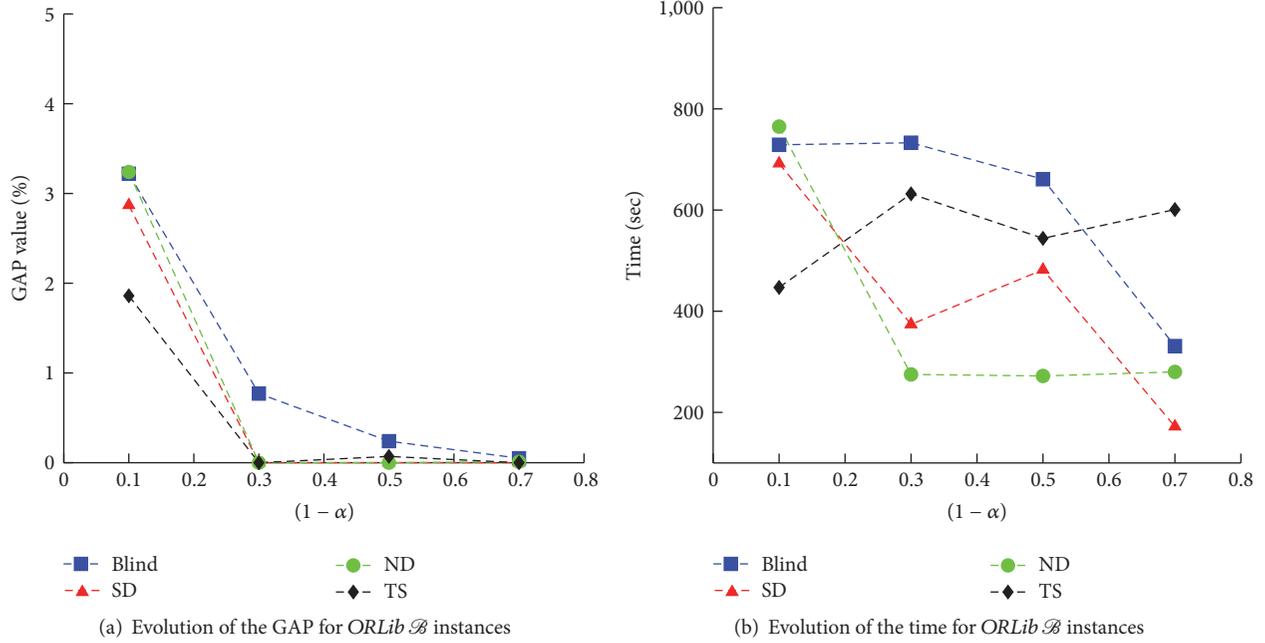


FIGURE 5: Average results obtained by the local search algorithms for each value of $(1 - \alpha)$ for ORLib B instances.

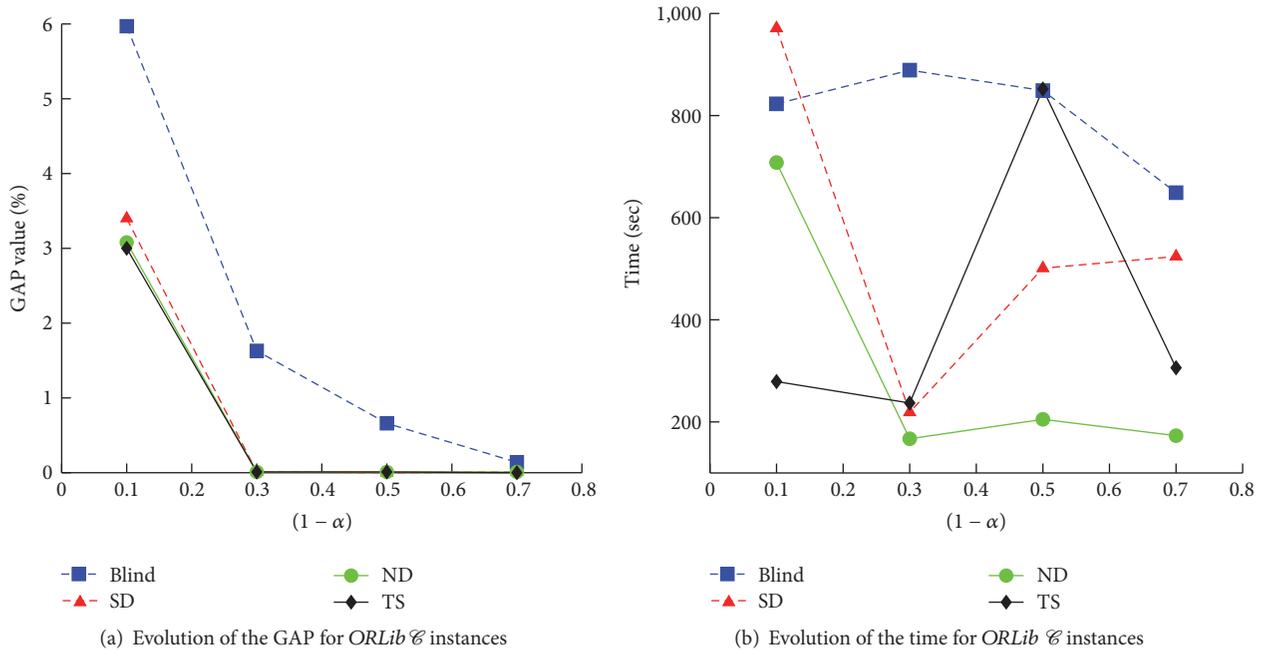


FIGURE 6: Average results obtained by the local search algorithms for each value of $(1 - \alpha)$ for ORLib C instances.

of facilities randomly so the problem has a feasible solution. However, this repairing process is completely random and, thus, algorithms performance is impaired.

Since the problems from the OR Library are only medium size instances, local-search-based matheuristics consistently find the optimal solution for almost all instances for $(1 - \alpha) = \{0.3, 0.5, 0.7\}$. In fact, even the blind algorithm finds solutions that are very close to the optimal ones when $(1 - \alpha) = 0.7$.

Figures 4(b), 5(b), and 6(b) show the time needed by our algorithms to find its best solution, as parameter $(1 - \alpha)$ increases. Unlike we expected, the time needed to find the best solution does not increase as the parameter $(1 - \alpha)$ gets larger. In fact, both algorithms, SD and blind, converge faster when $(1 - \alpha) = 0.7$, the larger value we tried in our experiments. This can be explained because of the problems features. We note that optimal solutions for all the

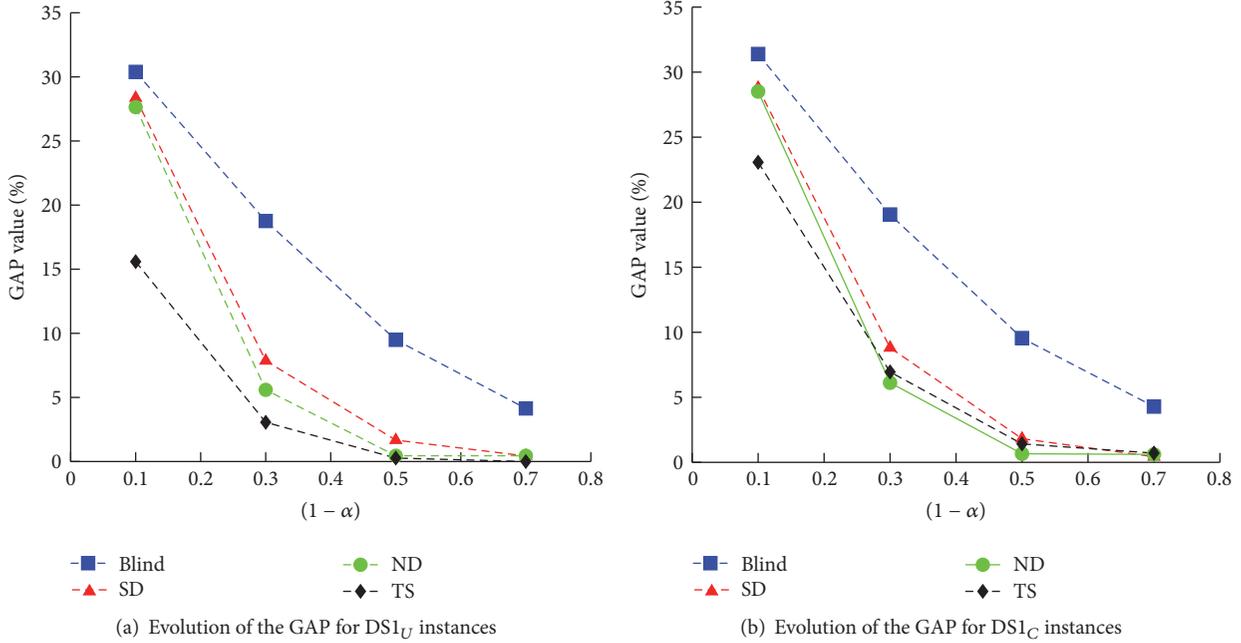


FIGURE 7: Average GAP values obtained by the local search algorithms for each value of $(1 - \alpha)$ for DS1 instances.

problems in the benchmark of the OR Library need not too many warehouses to be open. Thus, when a large portion of the potential warehouses are available (as for the $(1 - \alpha) = 0.7$ case) the algorithm is likely to find such optimal solutions in early iterations. In fact, for the vast majority of the experiments, optimal solution is found within the first 2 to 3 iterations.

We now move on the $DS1_U$ and the $DS1_C$ benchmarks. As we noted before, these sets of instances are much larger and harder to solve than the problems in the OR Library we discussed above. Further, as the optimal solutions of these instances do not require too many warehouses to be open, the repairing procedure applied to the OR Library instances for $(1 - \alpha) = 0.1$ does not have a great impact on the results. Thus, local-search-based matheuristics outperform the results obtained by the blind algorithm for all values of parameter α . Figures 7(a) and 7(b) show the GAP values obtained by the all four algorithms for instances $DS1_U$ and $DS1_C$, respectively.

Just as in the OR Library instances, as the parameter $(1-\alpha)$ gets larger, the GAP approximates to 0 for the all three local-search-based matheuristic proposed in this paper. While for the $DS1_U$ instances both the TS and the ND algorithms reach GAP values very close to 0 for $(1 - \alpha) = 0.5$ and $(1 - \alpha) = 0.7$, for the $DS1_C$ instances the best value obtained by the TS and the ND algorithms is 0.71 and 0.61, respectively, for $(1 - \alpha) = 0.7$. Notably, the ND algorithm performs slightly better than the TS for the $DS1_C$ instance when $(1 - \alpha) = 0.5$, obtaining a best average value of 0.66 against the 1.42 obtained by the TS algorithm. In spite of that, we can note that the differences between the GAP obtained by both the ND and the TS algorithms when using $(1 - \alpha) = 0.5$ and $(1-\alpha) = 0.7$ are negligible, just as in the OR Library instances.

Figures 8(a) and 8(b) shows the time needed by the all four algorithms to find their best solution for instances $DS1_U$ and $DS1_C$, respectively. As expected, as the parameter $(1 - \alpha)$ gets larger, longer times are needed by the algorithms to converge. This is important as one would like to use a value for $(1 - \alpha)$ such that minimum GAP values are reached within few minutes.

Results in Figures 7 and 8 show, clearly, that there is a compromise between the quality of the solution found by the algorithms and the time they need to do so. Further, we know that this compromise can be managed by adjusting the value of parameter $(1 - \alpha)$. For the case of instances $DS1_U$ and $DS1_C$, such a value should be between 0.4 and 0.6. Moreover, for the case of the instances of the OR Library, parameter $(1 - \alpha)$ should be set around 0.3. It is interesting to note that while the size of instances $DS1_U$ and $DS1_C$ is equivalent, instances of the OR Library are much smaller.

We can also note that instances that include clusters tend to take longer to converge. This is especially true as parameter $(1 - \alpha)$ gets larger. Further, for these instances, as the parameter $(1 - \alpha)$ gets larger, less iterations are performed by algorithms although each of these iterations takes much longer. As mentioned before, these should be taken into account when using population-based algorithms within the framework presented in this study as such kind of heuristic algorithms needs to perform several iterations before to converge to good quality solutions.

4. Conclusions and Future Work

In this paper we show the impact of parameter tuning on a local-search-based matheuristic framework for solving

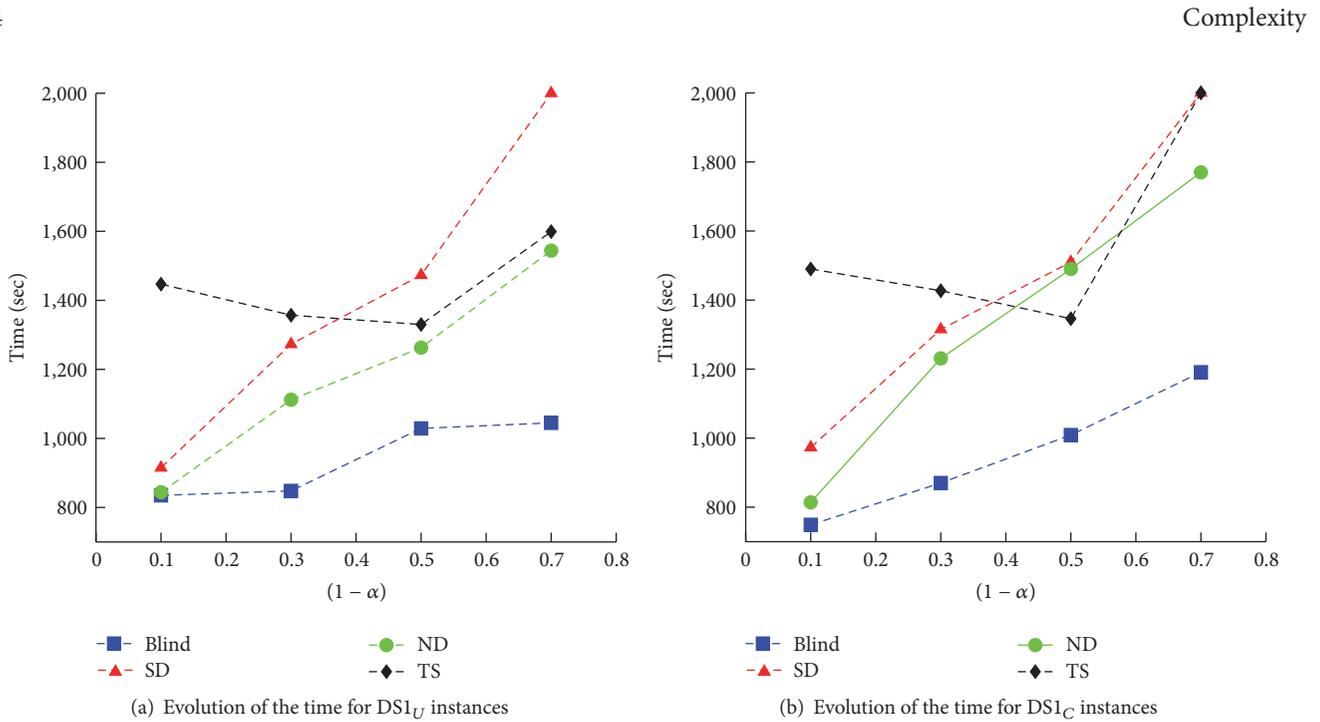


FIGURE 8: Average times needed by the local search algorithms for each value of $(1 - \alpha)$ for $DS1$ instances.

mixed integer (non)linear programming problems. In particular, matheuristics that combine local search methods and a MIP solver are tested. In this study, we focus on the size of the subproblem generated by the local search method that is passed on to the MIP solver. As expected, the size of the subproblems that are solved in turn by the matheuristic method has a big impact on the behaviour of the matheuristic and, consequently, on its obtained results: as the size of the subproblem increases (i.e., more integer/binary decision variables are considered) the results obtained by the MIP solver are closer to the optimal solution. The time required by the algorithms tested in this paper to find its best solution also increases as the subproblem gets larger. Further, as the subproblem gets larger, fewer iterations can be performed within the allowed time. This is important as other heuristics such as evolutionary algorithms and swarm intelligence, where many iterations are needed before converging to a good quality solution, might be not able to deal with large subproblems. We also note that the improvement in the GAP values after certain value of parameter $(1 - \alpha)$ is negligible and that this value depends to some extent on the size of the problem: for medium size instances such as the ones in the OR Library, parameter $(1 - \alpha)$ should be set to a value around 0.3 while for larger instances ($DS1_U$ and $DS1_C$) it should be set to a value between 0.4 and 0.6. The specific values will depend on both the accuracy level requested by users and the time available to perform the algorithm. Therefore, the challenge when designing matheuristic frameworks as the one presented in this paper is to find a value for parameter $(1 - \alpha)$ that allows the heuristic algorithm to perform as many iterations as needed and that provides a good compromise between solution quality and run times. Although the values provided here are only valid for the

problem and the algorithms considered in this paper, we think that the obtained results can be used as a guide by other researchers using similar frameworks and/or dealing with similar problems.

As a future work, strategies such as evolutionary algorithms and swarm intelligence will be tested within the matheuristic framework considering the results obtained in this study. We expect that intelligent methods such as the ones named before greatly improve the results obtained by the local search methods considered in this study. Moreover, the matheuristic framework used in this paper might also be applied to other MILP and MINLP problems such as, for instance, the beam angle optimisation problem in radiation therapy for cancer treatment.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Guillermo Cabrera-Guerrero wishes to acknowledge *Ingeniería 2030*, DI 039.416/2017 and CONICYT/FONDECYT/INICIACION/11170456 projects for partially supporting this research.

References

- [1] F. Neri and C. Cotta, "Memetic algorithms and memetic computing optimization: a literature review," *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.

- [2] Y.-s. Ong, M. H. Lim, and X. Chen, "Memetic Computation—Past, Present & Future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.
- [3] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.
- [4] V. Maniezzo, T. Stützle, and S. Voß, Eds., *Matheuristics - Hybridizing Metaheuristics and Mathematical Programming*, vol. 10 of *Annals of Information Systems*, Springer, 2010.
- [5] G. Guillermo Cabrera, E. Cabrera, R. Soto, L. J. M. Rubio, B. Crawford, and F. Paredes, "A hybrid approach using an artificial bee algorithm with mixed integer programming applied to a large-scale capacitated facility location problem," *Mathematical Problems in Engineering*, vol. 2012, Article ID 954249, 14 pages, 2012.
- [6] M. Caserta and S. Voß, "A math-heuristic Dantzig-Wolfe algorithm for capacitated lot sizing," *Annals of Mathematics and Artificial Intelligence*, vol. 69, no. 2, pp. 207–224, 2013.
- [7] L. C. Coelho, J.-F. Cordeau, and G. Laporte, "Heuristics for dynamic and stochastic inventory-routing," *Computers & Operations Research*, vol. A, no. 0, pp. 55–67, 2014.
- [8] C. Lagos, F. Paredes, S. Niklander, and E. Cabrera, "Solving a distribution network design problem by combining ant colony systems and lagrangian relaxation," *Studies in Informatics and Control*, vol. 24, no. 3, pp. 251–260, 2015.
- [9] B. Raa, W. Dullaert, and E.-H. Aghezzaf, "A matheuristic for aggregate production-distribution planning with mould sharing," *International Journal of Production Economics*, vol. 145, no. 1, pp. 29–37, 2013.
- [10] H. Allaoua, S. Borne, L. Létocart, and R. Wolfler Calvo, "A matheuristic approach for solving a home health care problem," *Electronic Notes in Discrete Mathematics*, vol. 41, pp. 471–478, 2013.
- [11] G. Cabrera-Guerrero, M. Ehrgott, A. J. Mason, and A. Raith, "A matheuristic approach to solve the multiobjective beam angle optimization problem in intensity-modulated radiation therapy," *International Transactions in Operational Research*, 2016.
- [12] Y. Li, D. Yao, J. Zheng, and J. Yao, "A modified genetic algorithm for the beam angle optimization problem in intensity-modulated radiotherapy planning," in *Artificial Evolution*, E.-G. Talbi, P. Liardet, P. Collet, E. Lutton, and M. Schoenauer, Eds., vol. 3871 of *Lecture Notes in Computer Science*, pp. 97–106, Springer, Berlin, Germany, 2006.
- [13] Y. Li, D. Yao, W. Chen, J. Zheng, and J. Yao, "Ant colony system for the beam angle optimization problem in radiotherapy planning: a preliminary study," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, volume 2, pp. 1532–1538, IEEE, Edinburgh, Scotland, UK, September 2005.
- [14] S. Hanafi, J. Lazić, N. Mladenović, C. Wilbaut, and I. Crévits, "New hybrid matheuristics for solving the multidimensional knapsack problem," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 6373, pp. 118–132, 2010.
- [15] E.-G. Talbi, "Combining metaheuristics with mathematical programming, constraint programming and machine learning," *4OR*, vol. 11, no. 2, pp. 101–150, 2013.
- [16] C. Archetti and M. G. Speranza, "A survey on matheuristics for routing problems," *EURO J. in Computer Optimization*, vol. 2, no. 4, pp. 223–246, 2014.
- [17] M. A. Boschetti, V. Maniezzo, M. Roffilli, and A. Bolufé Röhler, "Matheuristics: Optimization, simulation and control," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 5818, pp. 171–177, 2009.
- [18] C.-H. Chen and C.-J. Ting, "Combining lagrangian heuristic and ant colony system to solve the single source capacitated facility location problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 6, pp. 1099–1122, 2008.
- [19] Y. Li, J. Yao, and D. Yao, "Automatic beam angle selection in IMRT planning using genetic algorithm," *Physics in Medicine and Biology*, vol. 49, no. 10, pp. 1915–1932, 2004.
- [20] Y. Li, D. Yao, J. Yao, and W. Chen, "A particle swarm optimization algorithm for beam angle selection in intensity-modulated radiotherapy planning," *Physics in Medicine and Biology*, vol. 50, no. 15, pp. 3491–3514, 2005.
- [21] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Norwell, MA, USA, 1997.
- [22] M. Pirlot, "General local search methods," *European Journal of Operational Research*, vol. 92, no. 3, pp. 493–511, 1996.
- [23] C. Lagos, G. Guerrero, E. Cabrera et al., "A Matheuristic Approach Combining Local Search and Mathematical Programming," *Scientific Programming*, vol. 2016, Article ID 1506084, 2016.
- [24] V. Vails, M. A. Perez, and M. S. Quintanilla, "A tabu search approach to machine scheduling," *European Journal of Operational Research*, vol. 106, no. 2-3, pp. 277–300, 1998.
- [25] S. Hanafi and A. Freville, "An efficient tabu search approach for the 0-1 multidimensional knapsack problem," *European Journal of Operational Research*, vol. 106, no. 2-3, pp. 659–675, 1998.
- [26] J. Brandão and A. Mercer, "A tabu search algorithm for the multi-trip vehicle routing and scheduling problem," *European Journal of Operational Research*, vol. 100, no. 1, pp. 180–191, 1997.
- [27] K. S. Al-Sultan and M. A. Al-Fawzan, "A tabu search approach to the uncapacitated facility location problem," *Annals of Operations Research*, vol. 86, pp. 91–103, 1999.
- [28] P. B. Mirchandani and R. L. Francis, Eds., *Discrete Location Theory*, Wiley-Interscience, New York, NY, USA, 1990.
- [29] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.
- [30] J. G. Villegas, F. Palacios, and A. . Medaglia, "Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example," *Annals of Operations Research*, vol. 147, pp. 109–141, 2006.