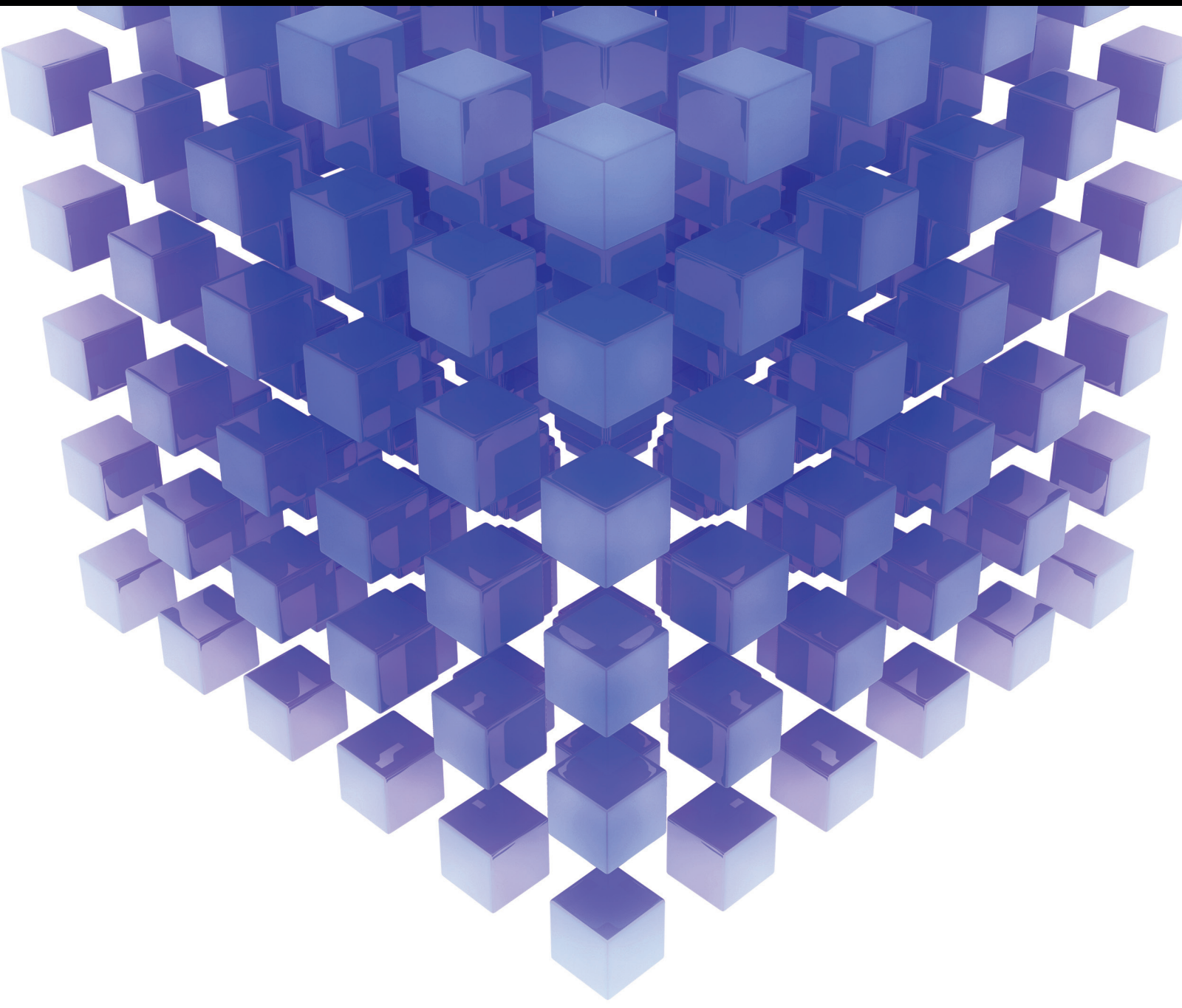
Mathematical Problems in Engineering

Optimization Theory, Methods, and Applications in Engineering 2014

Guest Editors: Jung-Fa Tsai, John Gunnar Carlsson, Dongdong Ge, Yi-Chung Hu, and Jianming Shi



Optimization Theory, Methods, and Applications in Engineering 2014

Mathematical Problems in Engineering

Optimization Theory, Methods, and Applications in Engineering 2014

Guest Editors: Jung-Fa Tsai, John Gunnar Carlsson, Dongdong Ge, Yi-Chung Hu, and Jianming Shi

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 345858, 3 pages
http://dx.doi.org/10.1155/2015/345858

Editorial

Optimization Theory, Methods, and Applications in Engineering 2014

Jung-Fa Tsai,[1] John Gunnar Carlsson,[2] Dongdong Ge,[3] Yi-Chung Hu,[4] and Jianming Shi[5]

[1]*Department of Business Management, National Taipei University of Technology, No. 1, Section 3, Chung-Hsiao East Road, Taipei 10608, Taiwan*
[2]*Program in Industrial and Systems Engineering, University of Minnesota, 111 Church Street SE, Minneapolis, MN 55455, USA*
[3]*Research Center for Management Science and Data Analytics, School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China*
[4]*Department of Business Administration, Chung Yuan Christian University, Chung-Li 320, Taiwan*
[5]*School of Management, Tokyo University of Science, 500 Shimokiyoku, Kuki, Saitama 346-8512, Japan*

Correspondence should be addressed to Jung-Fa Tsai; jftsai@ntut.edu.tw

Received 12 July 2015; Accepted 12 July 2015

Optimization theory and methods have been applied in various fields to handle various practical problems over the past few decades. Many approaches have been proposed to solve engineering optimization problems, and the methods can be divided into deterministic and heuristic ones. With the advances in computing techniques and the increasing reliance on optimization-based approaches in real applications, optimization techniques have continuously received attention from the practitioners and the researchers and many novel approaches have been proposed in the last few years. This special issue aims to present some recent developments in the area of optimization theory, methods, and applications in engineering. This special issue involves 17 original papers, selected by the editors and related to the various researches themes on optimization theory, methods, and applications in engineering. According to the characteristics of the accepted papers, the special issue is organized as the following three parts and each part is composed of several important papers to the part's scope.

Deterministic Optimization. W. Si et al. present a novel direction of arrival estimation method in compressed sensing. The proposed method is derived through transforming quadratically constrained linear programming into unconstrained convex optimization. Their method overcomes the drawback that l_1-norm is nondifferentiable when sparse sources are reconstructed by minimizing l_1-norm. The convergence rate and estimation performance of the proposed method can be significantly improved, since the steepest descent step and Barzilai-Borwein step are alternately used as the search step in the unconstrained convex optimization. Simulation results show the superior performance of the proposed method as compared with existing methods.

"Energy-Saving Generation Dispatch Using Minimum Cost Flow" by Z. Zhang and X. Cai uses a minimum cost flow method to solve a dispatch problem in order to minimize the consumption of coal in the dispatching of a thermal power system. The minimum cost flow problem, considering the loss of network flow, is known as a generalized network flow problem, which can be expressed as a quadratic programming problem in mathematics. Theoretical analysis and numerical results prove the correctness and effectiveness of the proposed method.

W. Wei, et al. propose a new technique of vehicle handling inverse dynamics which can evaluate the emergency collision avoidance performance. Based on the optimal control theory, the steering angle input and the traction/brake force imposed by a driver are the control variables; the minimum time required to complete the fitting biker line change is the control object. By using the improved direct multiple shooting method, the optimal control problem is converted

into a nonlinear programming problem that is then solved by means of the sequential quadratic programming. The simulation results show that the proposed method can solve the vehicle minimum time maneuver problem and can compare the maneuverability of two different vehicles that complete fitting biker line change with the minimum time, and the correctness of the model is verified through real vehicle test.

"Robust Control of the Knee Joint Angle of Paraplegic Patients considering Norm-Bounded Uncertainties" by N. M. dos Santos et al. presents a proposal for the knee position control design of paraplegic patients with functional electrical stimulation using control systems and considering norm-bounded uncertainties. A state-space representation of the knee joint model of the paraplegic patient with its nonlinearity is demonstrated and the use of linear matrix inequalities in control systems with norm-bounded uncertainties for asymptotic stability is analyzed.

S. Choi and K. Park study a few dynamic risk-averse inventory models using additive utility functions. They add Markovian behavior of purchasing costs in the models and provide the problem formulations with finite and infinite MDP (Markovian Decision Process) problems. For finite time models, they first prove (joint) concavity of the model for each state and obtain a (modified) base-stock optimal policy. Then, they conduct comparative static analysis for model parameters and derive monotone properties to the optimal solutions. For infinite time models, they show the existence of stationary base-stock optimal policies and the inheritance of the monotone properties proven at the finite time models.

X. Xian-hao et al. investigate the capacity allocation policies of a third-party warehouse center, which supplies several different level services on different prices with fixed capacity, on revenue management perspective. For the single-period situation, they use three different robust methods, absolute robust, deviation robust, and relative robust, to maximize the whole revenue. For the multiperiod situation, as the demand is uncertain, they propose a stochastic model for the multiperiod revenue management problem of the warehouse. A novel robust optimization technique is applied to maximize the whole revenue in the proposed model. Then some numerical examples are given to verify the practical applicability of their method.

The paper "Equilibrium Customer Strategies in the Queue with Threshold Policy and Setup Times" by P. Chen et al. considers the equilibrium behavior of customers in the M/M/1 queue with N policy and setup times and addresses the equilibrium threshold strategies for the fully observable case and mixed strategies for the fully unobservable case, respectively. This study also gets various performance measures of the system and investigates some numerical examples of system size, social benefit, and expected cost function per unit time for the two different cases under equilibrium customer strategies.

Heuristic Algorithms. The paper "Metaheuristic Approaches for Solving Truck and Trailer Routing Problems with Stochastic Demands: A Case Study in Dairy Industry" by S. Mirmohammadsadeghi and S. Ahmed aims at modeling the truck and trailer routing problem with stochastic demand constraints to bring the truck and trailer routing problem model closer to a reality. The model is solved in a reasonable timeframe using data from a large dairy service by administering the multipoint simulated annealing, memetic algorithm, and tabu search. A sizeable number of customers whose demands follow the Poisson probability distribution are considered to model and solve the problem. Experimental results indicate that the average results obtained by the proposed memetic algorithm (MA) are improved about 1.27 and 0.78 percent comparing with tabu search and multipoint simulated annealing, respectively. However, the differences between the results are insignificant. Consequently, the algorithms are found to be appropriate and relevant for solving the real-world truck and trailer routing problem with stochastic demands.

I. Hwang et al. propose a new heuristic that identifies clusters in the population of partitions which have already been optimized locally. The clusters must be created to seed the memetic algorithms, and as the memetic algorithm runs it makes the beneficial cluster moves. The research results on standard benchmark graphs show significant reductions in cut size and get better solutions than those in the literature for some cases.

"Maintaining Track Continuity for Extended Targets Using Gaussian-Mixture Probability Hypothesis Density Filter" by Y. Han et al. presents a multiextended target tracker based on the extended target Gaussian-mixture probability hypothesis density filter to provide the tracks of the extended targets in maintaining the track continuity for the extended targets. To improve the performance of the extended target tracker, the study also proposes a mixture partitioning algorithm for resolving the identities of the extended targets in close proximity. The simulation results show that the proposed tracker achieves the less error of the position estimates and decreases the probability of incorrect label assignments.

"A Variable Depth Search Algorithm for Binary Constraint Satisfaction Problems" by N. Bouhmala presents a fast metaheuristic for solving binary constraint satisfaction problems. The method can be classified as a variable-depth search metaheuristic combining a greedy local search using a self-adaptive weighting strategy on the constraint weights. Experiments are conducted on hard random constraint satisfaction problems enjoying several features that make them of a great theoretical and practical interest. The results show that the proposed metaheuristic is capable of solving hard unsolved problems that still remain a challenge for both complete and incomplete methods. In addition, the proposed metaheuristic is remarkably faster than all existing solvers when tested on previously solved instances. Finally, its distinctive feature contrary to other metaheuristics is the absence of parameter tuning making it highly suitable in practical scenarios.

The paper "An Effective Hybrid of Bees Algorithm and Differential Evolution Algorithm in Data Clustering" by M. B. Bonab et al. is to minimize the dissimilarity of all points of a cluster, from gravity center of the cluster with respect to capacity constraints in each cluster, such that

each element is allocated to only one cluster. This article presents a new hybrid algorithm based on cluster center initialization algorithm (CCIA), bees algorithm (BA), and differential evolution (DE), known as CCIA-BADE-K, aiming at finding the best cluster center. The evaluation results of the proposed algorithm and its comparison with other alternative algorithms in the literature confirm its superior performance and higher efficiency.

P. Liang et al. consider an unrelated parallel machine scheduling problem with energy consumption and total tardiness. They establish a mathematical model for this problem and then present an ant optimization algorithm based on ATC heuristic rule (ATC-ACO). Furthermore, the optimal parameters of the proposed algorithm are defined via Taguchi methods for generating test data. Finally, the results of comparative experiments indicate the proposed ATC-ACO algorithm has better performance on minimizing energy consumption as well as total tardiness and the modified ATC heuristic rule is more effective on reducing energy consumption.

Information System and Multicriteria Decision-Making. "A Real-Time Pothole Detection Approach for Intelligent Transportation System" by H.-W. Wang et al. proposes a pothole detection method based on the mobile sensing. The accelerometer data is normalized by Euler angle computation and is adopted into the pothole detection algorithm to obtain the pothole information. Moreover, the spatial interpolation method is used to reduce the location errors from global positioning system data. The experimental results show that the proposed approach can precisely detect potholes without false-positives, and the higher accuracy is performed by the proposed approach. Therefore, the proposed real-time pothole detection approach can be used to improve the safety of traffic for intelligent transportation system.

The paper "Parallel Control to Fragments of a Cylindrical Structure Driven by Explosive inside" by W. Chen et al. presents a new design method for the shell shape. To facilitate the establishment of the numerical model and the machining for relative experiments, the mathematical description of the theoretical calculated generatrix of the shell is simplified. Based on the theoretical design and plentiful simulation data, the relationships between the size of the parallel fragmentation structure and the optimized curvature radius of the shell are expressed by an equation. The equation is validated by numerical means and can be a reliable reference to the design of the parallel fragmentation structure.

"To Make Good Decision: A Group DSS for Multiple Criteria Alternative Rank and Selection" by C.-S. Wang et al. proposes a group decision support system architecture named hybrid decision-making support model (HDMSM), integrating four decision approaches (Delphi, DEMATEL, ANP, and MDS) to help the decision maker in ranking and selecting appropriate alternatives. The HDMSM consists of five steps, namely, criteria identification, criteria correlation calculation, criteria evaluation, critical criteria selection, and alternative rank and comparison. According to the case study demonstration, the proposed HDMSM can enable a group of decision makers to implement the MCDM effectively and help them to analyze the relation and degree of mutual influence among different evaluation factors.

The paper "Configuration, Deployment, and Scheduling Models for Management and Optimization of Patrol Services" by B. Yang et al. presents a decision support system (DSS) and its models for patrol service center (PSC). This work contributes to the literature on patrol services and network optimization problems in the following aspects: based on a series of models, a DSS framework is designed for PSCs; the models are formulated for resource management and scheduling upon geography information system; coordination strategies among close PSCs are incorporated into decision models. These features are examined in integration manners. The assessment criteria and optimization models studied in the paper are beneficial for building DSSs for PSC.

Jung-Fa Tsai
John Gunnar Carlsson
Dongdong Ge
Yi-Chung Hu
Jianming Shi

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 494019, 14 pages
http://dx.doi.org/10.1155/2015/494019

Research Article

Metaheuristic Approaches for Solving Truck and Trailer Routing Problems with Stochastic Demands: A Case Study in Dairy Industry

Seyedmehdi Mirmohammadsadeghi and Shamsuddin Ahmed

Manufacturing System Integration, Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, Kuala Lumpur 50603, Malaysia

Correspondence should be addressed to Seyedmehdi Mirmohammadsadeghi; mehdi_mirmohammadsadeghi@yahoo.com

Received 18 July 2014; Revised 13 November 2014; Accepted 18 December 2014

Academic Editor: Yi-Chung Hu

Manufacturers and service providers often encounter stochastic demand scenarios. Researchers have, thus far, considered the deterministic truck and trailer routing problem (TTRP) that cannot address ubiquitous demand uncertainties and/or other complexities. The purpose of this study is to model the TTRP with stochastic demand (TTRPSD) constraints to bring the TTRP model closer to a reality. The model is solved in a reasonable timeframe using data from a large dairy service by administering the multipoint simulated annealing (M-SA), memetic algorithm (MA), and tabu search (TS). A sizeable number of customers whose demands follow the Poisson probability distribution are considered to model and solve the problem. To make the solutions relevant, first, 21 special TTRPSD benchmark instances are modified for this case and then these benchmarks are used in order to increase the validity and efficiency of the aforementioned algorithms and to show the consistency of the results. Also, the solutions have been tested using sensitivity analysis to understand the effects of the parameters and to make a comparison between the best results obtained by three algorithms and sensitivity analysis. Since the differences between the results are insignificant, the algorithms are found to be appropriate and relevant for solving real-world TTRPSD problem.

1. Introduction

These days, complex customer demands are required to be satisfied by many companies. Therefore, a large number of companies are trying to achieve a high level of reliability, flexibility, and agility for different demands. As a result, supply chain management (SCM) has become a thought-provoking subject for various companies, seeking for a way out of efficiently improving their customer satisfaction. In a way, according to the position and role, supply chain is categorized into three classes; the outbound, intracompany, and inbound supply chain. As the network of supplies begins at the inbound supply chain, the role of this group is transporting the semifinished products or the raw materials to the site of manufacturing. The main concern of the intracompany supply chain, as the intermediary part, is with the flow of material in the site of manufacturing. Finally, the outbound supply chain is concerned with the delivery of final products to the customers [1]. The inventory allocation and transportation are strongly considered in the outbound supply chain for minimizing the cost and satisfying the customers' demands. One significant part of the supply chain management is providing the services or/and goods from a supply point to different destinations, which are geographically distributed with significant implications of economics. Aside from the cost of purchasing the goods, on the average and compared to the other relative activities, a higher percentage of the costs of logistics are absorbed by transportation. Therefore, efficiency improvement through the maximum usage of the necessities of transportation and decreasing the costs of transportation along with the improvement of services for customers are the frequent and significant decision analysis problems [2].

Customers, warehouses, manufacturers, and suppliers are the main elements of a supply chain (SC), carrying the goods

from the upstream to the downstream links of the chain. In a supply chain, there are four main business functions to be performed: purchasing, manufacturing, inventory, and distribution. The function of distribution includes two activities: the shipment of finished products from the companies to the locations of demand, and transportation of parts or raw materials from the suppliers to the companies [3].

In order to manage a supply chain, a large number of business processes need to be carried out and many decisions are required to be made. Particular design versions of these general supply systems and inventory planning problems have been studied for a long time. It is pretty obvious that these main supply chain problems are greatly related. As the time goes by, more companies are awakened about their supply chain performance and how important it is that they improve this performance. They also have become aware of the competitive advantage of distribution operations, inventory, integration, and coordination of supplies. One of the main problems in supply chain management and logistics is the routing of a series of vehicles, which are assigned to transport goods from a warehouse to the customers or/and retailers. Since goods are hardly ever produced and consumed in one particular place, transportation is considered as a significant factor in the supply chain management.

In this study, the supply chain is considered in terms of transportation and distribution. Due to the complexity, a large number of realistic solutions are disregarded. Any company in the world currently faces with a number of challenges in serving their customers. Transportation cost is considered to be the largest logistics expense for a vast number of firms and companies. Transportation is the area where costs should be diminished. This is a very bearing question, how servicing and manufacturing companies could successfully diminish transportation expenses without overshadowing customers' satisfaction [4].

It is a widely accepted principle that firms aiming to serve the customers scattered in a vast area should possess a servicing plan if they do not want to waste time and money. One of the best approaches to work out the arising problematic issues is to apply a special and unique method under the title of vehicle routing problem (VRP). Also, the truck and trailer routing problem (TTRP) is one of the widely studied and most important combinatorial optimization problems in VRP domain and because of its natural complications and efficacies in a large number of real world and supply chain management applications, it is finding its place in the transportation logistic.

This paper discusses a real case study under TTRP in which demands are stochastic (TTRPSD) in nature. This work is an advancement of the well-known deterministic TTRP. TTRP is a variant of the vehicle routing problem (VRP). It is related to transporting manufacturing goods within a plant or between factory floors and delivering products to intended markets or customers. Indeed, VRP has been known as one of the most studied combinatorial optimization problems in the past few decades. It covers certain areas in practice and considers complexities to a reasonable extent such as stochastic VRP, multi depot VRP and TTRP [5–8]. Dantzig and Ramser [8] defined VRP as a generalized solution based on Travelling Salesman Problem (TSP) in 1959. After that many researchers have extended this concept to make it useful in diverse areas. During the last two decades, some constraints such as time window, travel and service time were added to the VRP. Due to some other practical issues, such as narrow roads and bridges in village or government restrictions, maneuvering a complete vehicle sometime appears to be difficult - the traditional VRP approach is found inadequate and these issues are considered in TTRP model. In general, TTRP is more extensive than VRP and can cover more real life aspects since some limitations in VRP as mentioned above can be considered in TTRP.

In TTRP, the customers may be serviced either by a single truck or complete vehicle (truck with a trailer). This feature is usually ignored in VRP. However, because of some obstacles that appear in real-life situations, such as road conditions, market locations, government regulations or limited space to manoeuver at the customer's site, only a single truck may be needed to serve a few destinations and/or customers. These constraints are obvious in many practical situations [9–11]. Several researchers have so far contributed in this area. One instance is that of Gerdessen's work on VRP with a trailer in 1996 [12]. He demonstrated two real applications pertaining to TTRP. In one case it was the distribution of dairy products in cities where the distributor faced heavy traffic. Due to the fact that maneuvering a complete vehicle (a truck with a trailer) was difficult in some areas, and that some streets had traffic restrictions not allowing trailers to enter, the trailers were often parked in parking spots from which customers were serviced. The second case was the distribution of animal feed components to farmers. Because most villages have narrow roads or small bridges, different kinds of vehicles were required to deliver the feed to farmers, one of which was called a double bottom, consisting of a truck and a trailer. While the truck made deliveries using subtours (some parts of the tour) the trailer parked in a parking area. Gerdessen showed the necessity of considering TTRP by demonstrating real applications.

In practical situations, a dispatcher may not know the exact demands in advance. Therefore, the company may face the problem of delivering products to customers with random demands. Consequently, unexpected extra costs might be imposed on the company. These issues can be considered in vehicle routing problems with stochastic demands (VRPSD). Moreover, when facing the limitations and restrictions mentioned above, VRPSD cannot cover these issues and needs to consider TTRP with stochastic demands (TTRPSD). The literature survey revealed no TTRP solution with stochastic parameters. Few articles were published on TTRP with deterministic parameters. However, papers published on stochastic VRP are simply large in number. These concepts need to be considered together in order to formulate TTRPSD. Therefore, this paper classifies the relevant papers into two groups—standard TTRP papers and VRP with stochastic demands. To solve TTRPSD, it appears that its solution is computationally more difficult than VRPSD. Indeed, VRP itself is one of the most difficult combinatorial optimization modelling problems. It is generally framed and solved by heuristics approaches [13–15]. To formulate and solve

TTRPSD, one can make an effort to reduce it to VRPSD. As VRPSD is also a complex type of combinatorial optimization problem, it can be solved by heuristics approaches [7, 10, 16]. The purpose of TTRPSD is to design a set of routes to cover all customers and optimize the total costs or distance that will satisfy the relevant constraints.

Standard TTRP was first proposed by Chao [7]. Scheuerer applied 0-1 integer programming formulation to solve TTRP [16]. Chao and Scheuerer used a two-phase approach to solve TTRP [7, 16]. They used heuristics to construct the initial TTRP solution in the first phase. In the second phase, the Tabu search algorithm was used to improve the initial solution. Chao followed Fisher and Jaikumar's construction to solve VRP [17]. Scheuerer [16] used Chao's model [7] and improved it by using two-construction heuristics, T-Cluster and T-Sweep, and applied a new Tabu search improvement algorithm to solve TTRP. Lin et al. [9] introduced simulated annealing to solve TTRP and obtained 17 best solutions to the 21 benchmarked TTRPs. Then they applied time window constraints to the TTRP solution for the first time to bring the model closer to reality [18]. Villegas et al. [19] considered a single TTRP with satellite depots (STTRPSD). Variable neighbourhood descent (VND) and greedy randomized adaptive search procedures (GRASP) were proposed by them for solving TTRP. In addition, they applied the GRASP/VND algorithm for multidepot VRPs and improved the previous analysis. Villegas et al. [20] improved this solution by considering a hybrid algorithm based on the GRASP/VND and a path relinking (PR) algorithm and proved that the performance of this hybrid algorithm exceeds that of GRASP/VND. Finally, Villegas et al. [11] proposed a new hybrid algorithm by considering GRASP and an iterated local search (ILS) and found a new solution for benchmarking, which was considered by Lin et al. for the first time [9]. Derigs et al. proposed TTRP without transferring load between truck and trailer for the first time [10]. A hybrid algorithm was applied to solve the TTRP by considering the large neighbourhood search (LNS) and local search (LS). In addition, time window constraints were also considered for each customer to bring the model closer to reality. Tillman introduced stochastic demand in capacitated VRP (CVRP) for the first time [21, 22]. In his study, the multidepot VRP was considered in which the demands are stochastic with Poisson distribution. Bertsimas demonstrated critical evidence for the stochastic VRP and compared reoptimization and *a priori* strategies [23]. Gendreau et al. [24] presented a Tabu search algorithm for solving VRP with stochastic demands and customers' demands following a known distribution and customers were presented with a probability. The integer L-shaped method for CVRP with normal and Poisson demands was improved by Laporte et al. [25]. Lei et al. [26] extended CVRP with stochastic demand (CVRPSD), and time window constraints were added to the vertices. They solved the vehicle routing problem with stochastic demands and time windows (VRPSDTW) by considering discrete and continuous distributions for stochastic demands.

In this study, the multipoint simulated anneling (M-SA), memetic algorithm (MA) and tabu search (TS) algorithms are applied in solving the TTRPSD. Multipoint SA is a local search heuristic method, which uses a few initial solutions instead of one solution and generates better solutions in the process in order to find the best solutions. The MA is a hybrid algorithm which combines the population and LS procedures to find the best solutions. Tabu search is an iterative local search heuristics. A tabu mechanism is put in place to prevent the process from cycling over a sequence of solutions. This paper demonstrates these algorithms that are efficient for solving TTRPSD. The new real case study has been considered for this purpose. This problem has been solved by these algorithms and the results have been compared with each other. In addition, sensitivity analysis is conducted to validate the results.

This paper is organized as follows. Section 2 provides the formulation of the TTRPSD. The detailed implementation of metaheuristic algorithms for solving the TTRPSD is described in Section 3. The computational study is reported in Section 4. The paper is concluded in Section 5. In addition, some further researches are recommended in this section.

2. Formulation of Truck and Trailer Routing Problem with Stochastic Demand

TTRPSD is defined as an undirected graph $G = (V, E)$, where the set of vertices is $V = \{v_0, v_1, v_2, \ldots, v_n\}$ and $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ is the set of edges. The central depot is represented by "v_0" and the other vertices in $V \setminus \{v_0\}$ correspond to customers. Each vertex v_i is associated with a stochastic and nonnegative demand ξ_i. They can be split and are unknown until the vehicle arrives at the vertex. A customer type t_i is available for all customers, where $t_i = 1$ shows that customer v_i is a truck customer (TC) and can be serviced only by a single truck. If $t_i = 0$, a customer i is a vehicle customer (VC) and it can be serviced by a single truck or a complete vehicle (truck pulling a trailer). $C = (c(v_i, v_j))$ is a symmetric travel cost, which is defined on E. It is assumed that all vehicles are the same and have a unit speed, so the travel cost is equal to the Euclidean distance between v_i and v_j. Say a fleet of m_k and m_r, trucks and trailers, respectively, is available. However, some trucks and trailers may not be used at any time in the TTRPSD solution. Without losing generality, it is assumed that $m_k \geq m_r$, as in Chao [7], Scheuerer [16] and Lin et al. [9]. All trucks have the same capacity Q_k, and all trailers also have the same capacity Q_r, where Q_k and Q_r are different.

Three types of route are available in TTRPSD as follows: (1) a pure travel route (PTR) which can be travelled by only a single truck; (2) a pure vehicle route without any subtours (PVR), which can only be travelled by a complete vehicle; (3) complete vehicle route (CVR) which consists of a main tour and at least one subtour. A subtour starts and finished at the same VC (v_r) (the trailer is parked in a parking place which is called the root) and it can be travelled only by a single truck; however, it should be serviced by a complete vehicle in the main tour. It is assumed that shifting demand loads between the truck and trailer is possible at the parking places. In addition, it should be mentioned that there may be a lack of capacity in serving the customers since the customers' demands are stochastic. Therefore, the vehicle must return to

the depot or to the parking place (while the truck is delivering on the subtours) and refill to capacity to serve the remaining customers. This is the so-called failure in the route.

The TTRPSD can be cast as stochastic programming. Two main solution concepts for solving the aforesaid types of TTRPSD can be discerned from stochastic programming [26]. The first is known as chance constrained programming (CCP). In CCP, the problem can be solved by imposing a constraint ensuring that the probability of route failure is bounded by some parameters, such as time limitation and service time [26, 27]. This concept attempts to convert stochastic parameters to equivalent deterministic values. For instance, the TTRPSD can be converted to an equivalent deterministic program. Stewart and Golden [28] and Laporte et al. [29] demonstrated this transformation by considering the statistical relationships among the parameters. The second concept is stochastic programming with recourse (SPR). Two main solution strategies are available under SPR. The first is known as *a priori* optimization [23, 30–32] and the second is reoptimization [33, 34]. In an *a priori* optimization solution, the set of tours and subtours is constructed in the first stage. Recourse actions considering random variables are then revealed. The common recourse policy in TTRPSD is as follows: if the cumulative demand exceeds the vehicle capacity on the main tour (it means that the cumulative demand exceeds the summation of capacities of truck and trailer on the main tour up to a vertex), the vehicle returns to the depot to unload, fills the capacity (truck and trailer) and then comes back to the last visited vertex. However, if the vehicle capacity is reached exactly for any vertex on the main tour (it means that the cumulative demand is equal to the summation of capacities of truck and trailer on the main tour up to a vertex), the vehicle should return to the depot, fills the capacity and then start servicing from the next vertex along its route. But if the cumulative demand exceeds the truck capacity on the subtour, the vehicle should return to its root to upload, use the remaining trailer capacity and return to the last visited vertex on the subtour. In addition, if the truck capacity exactly reached the customer demand for any vertex on the subtour (it means that the cumulative demand on the subtour is equal to the truck capacity up to a vertex), the truck returns to the root, using the remaining trailer capacity to fill the truck and comes back to the next vertex on the subtour.

The TTRPSD consists of designing the first-stage routes, including the truck route, vehicle route and complete route, satisfying all constraints and if any failure occurs, a recourse action is applied. The purpose is to minimize the sum of the total distance of the first-stage routes and the expected recourse costs.

2.1. The Expected Cost of a Solution. Since all trucks and trailers in their respective groups are the same, each having its own features in terms of capacity and associated unit speed, the cost is associated with the distance travelled. Initially, the expected recourse cost (distance) is computed under some assumptions. Then the total cost is computed and the recourse costs will be computed based on these assumptions.

It should be mentioned that all demands are independent random variables with known distributions. Also each demand ξ_i is a nonnegative random variable that never exceeds the truck capacity. In addition, a maximum of one return to the root is possible on any subtour (ST) and a maximum of one return to the depot can be allowed on any main tour (in a complete route), truck or vehicle route. Because more than one failure is impossible on any kind of route r_k

$$P\left\{\sum_{v_i\in r_k}\xi_i < 2\left(Q_k + Q_r\right)\right\} \cong 1 \quad \text{If } r_k \in \text{PVR, CVR,}$$
$$P\left\{\sum_{v_i\in r_k}\xi_i < 2\left(Q_k\right)\right\} \cong 1 \quad \text{If } r_k \in \text{PTR, ST.} \tag{1}$$

This failure means that the demand of a customer cannot be satisfied while serving the customers since the vehicle does not have enough capacity to serve the respective customer and has to come back to the depot (or parking place) and fill the capacity and return to the respective customer to serve it completely. This is to ensure that this failure may occur a maximum once. Therefore, the cumulative demand of the customers must be less than twice the route capacity in a worst case.

2.2. Mathematical Estimation of the Total Expected Cost. It should be considered that TTRPSD has three different types of route. Each route is planned in the first stage of the solution $R = (r_1, r_2, \ldots, r_m)$, where $r_k = (v_0^k = v_0, v_1^k, \ldots, v_{n_k}^k, v_{n_k+1}^k = v_0)$. The final objective function $T(R)$ is the sum of two costs: $\Phi(R)$ and $R(R)$, where $\Phi(R)$ is the total distance of the first-stage routes, and $R(R)$ is the expected recourse cost. So, the objective function is,

$$T(R) = \Phi(R) + R(R). \tag{2}$$

The computation of $\Phi(R)$ is not difficult. To compute $\Phi(R)$, the cost of each route should be found and the costs (total distances) of all routes, which are planned in the first stage, should be computed. The estimations of $\theta(R)$ is complicated and it is shown in the next section. First, the probability of failure should be computed. ξ_i^k and X_i^k are the demand from customer v_i and the cumulative demand up to customer v_i in a route r_k.

The probability of the cumulative demand up to the customer v_i^k, on a route r_k, can be calculated by the following recursive equation [24], where $P_{\xi_0^k}(t)$ is a boundary condition for this equation and $P_{X_i^k}(t) = P\{X_i^k = t\}$

$$P_{X_i^k}(t) = \sum_{d=0}^{t} P_{X_{i-1}^k}(t-d)\, P_{\xi_i}(d). \tag{3}$$

For instance, consider $P_{X_0^k}(t) = P_{\xi_0^k}(t), P_{X_4^k}(3) = P_{X_3^k}(3)P_{\xi_i}(0) + P_{X_3^k}(2)P_{\xi_i}(1) + P_{X_3^k}(1)P_{\xi_i}(2) + P_{X_3^k}(0)P_{\xi_i}(3)$. This means that the cumulative demand up to customer 4 can be three if the cumulative demand up to customer 3 is three and the demand of customer 4 is zero, or the cumulative demand up to customer 3 is two and the demand of customer 4 is one,

or the cumulative demand up to customer 3 is one and the demand of customer 4 is two, or the cumulative demand up to customer 3 is zero and the demand of customer 4 is three. Therefore, the probability of these cases should be summed up to calculate the total cumulative demand up to customer 4.

The failure occurred at vertex v_i^k on a route r_k as follows. If failure occurred in a pure truck route or ST, then $X_{i-1}^k < Q_k$ and $X_i^k \geq Q_k$. However, $X_{i-1}^k < Q_k + Q_r$ and $X_i^k \geq Q_k + Q_r$, if failure occurred in a pure vehicle route or a main tour. Then depending on the type of route, the probability P_i^k can be computed. The probability of route failure P_i^k at customer v_i^k on a route r_k can be computed [26] as

$$P_i^k = \begin{cases} \sum_{d=0}^{Q_k-1} P_{X_{i-1}^k}(t) - \sum_{d=0}^{Q_k-1} P_{X_i^k}(t) \\ \qquad \text{if failure occurred in PTR or sub tour,} \\ \sum_{d=0}^{Q_k+Q_r-1} P_{X_{i-1}^k}(t) - \sum_{d=0}^{Q_k+Q_r-1} P_{X_i^k}(t) \\ \qquad \text{if failure occurred in PVR, main tour,} \\ 0 \qquad \text{other wise.} \end{cases} \tag{4}$$

For example, if the failure occurs on the main tour of CVR the probability P_i^k can be computed as

$$\begin{aligned} P_i^k &= P\left\{X_{i-1}^k < Q_k + Q_r,\ X_i^k \geq Q_k + Q_r\right\} \\ &= 1 - P\left\{X_{i-1}^k \geq Q_k + Q_r\right\} - P\left\{X_i^k < Q_k + Q_r\right\} \\ &= P\left\{X_{i-1}^k < Q_k + Q_r\right\} - P\left\{X_i^k < Q_k + Q_r\right\} \\ &= \sum_{l=0}^{Q_k+Q_r-1} P_{X_{i-1}^k}(t) - \sum_{l=0}^{Q_k+Q_r-1} P_{X_i^k}(t). \end{aligned} \tag{5}$$

2.3. The Expected Recourse Cost. Failure can happen in different situations. First, failure may occur in the PTR, PVR or main-route on the CVR; second failure may occur in an ST on the CVR. The recourse cost can be computed according to four failure types as follows.

(1) Type 1: $H_i^k = c(v_i^k, v_r^k) + c(v_r^k, v_{i+1}^k) - c(v_i^k, v_{i+1}^k)$, if $X_i^k + \xi_i^k = Q_k$, if failure occurred in the PTR or ST.

(2) Type 2: $G_i^k = 2c(v_i^k, v_r^k)$, if $X_i^k + \xi_i^k > Q_k$, if failure occurred in the PTR or ST.

(3) Type 3: $F_i^k = c(v_i^k, v_0^k) + c(v_0^k, v_{i+1}^k) - c(v_i^k, v_{i+1}^k)$, if $X_i^k + \xi_i^k = Q_k + Q_r$, if failure occurred in the PVR, main tour.

(4) Type 4: $E_i^k = 2c(v_i^k, v_0^k)$, if $X_i^k + \xi_i^k > Q_k + Q_r$, if failure occurred in the PVR, main tour.

For instance, if the failure is of the first type, it means that after serving customer v_i^k, the vehicle has to return to the root of the route if the vehicle is in ST, otherwise it must return to the depot, then continue serving customers from the next vertex v_{i+1}^k. If the second failure type occurs, after serving customer v_i^k, the single truck must return to the depot or to the root of its route and again proceed to vertex v_i^k. The third and fourth types of the failure are the same as the first and second one.

The expected cost of recourse for route r_k can be computed as

$$\begin{aligned} E[\theta(r_k)] = \sum_{i=2}^{n_k} \Big(& A_i^k\left(H_i^k\right) + B_i^k\left(G_i^k\right) \\ & + C_i^k\left(F_i^k\right) + D_i^k\left(E_i^k\right)\Big), \end{aligned} \tag{6}$$

where $A_i^k, B_i^k, C_i^k, D_i^k$ is the probability that first, second, third, fourth types of failure are occurring respectively. Considering (3) and (4), the two terms can be computed as

$$A_i^k = \begin{cases} \sum_{l=1}^{Q_k} P_{\xi_i^k}(l)\, P_{X_{i-1}^k}(Q_k - l) \\ \qquad \text{if failure occurred in PTR or sub tour,} \\ 0 \qquad \text{otherwise,} \end{cases}$$

$$B_i^k = \begin{cases} \sum_{l=1}^{Q_k-1} P_{X_{i-1}^k}(t) - \sum_{l=1}^{Q_k-1} P_{X_i^k}(t) - A_i^k \\ \qquad \text{if failure occurred in PTR or sub tour,} \\ 0 \qquad \text{otherwise,} \end{cases}$$

$$C_i^k = \begin{cases} \sum_{l=1}^{Q_k+Q_r} P_{\xi_i^k}(l)\, P_{X_{i-1}^k}(Q_k + Q_r - l) \\ \qquad \text{if failure occurred in PVR, main tour,} \\ 0 \qquad \text{otherwise,} \end{cases}$$

$$D_i^k = \begin{cases} \sum_{l=1}^{Q_k+Q_r-1} P_{X_{i-1}^k}(t) - \sum_{l=1}^{Q_k+Q_r-1} P_{X_i^k}(t) - C_i^k \\ \qquad \text{if failure occurred in PVR, main tour,} \\ 0 \qquad \text{otherwise.} \end{cases} \tag{7}$$

To recognize the failure types and their probabilities, it should be mentioned that the expected cost of a recourse is an extra travel cost. Therefore, four failure types and four recourse actions have been considered for this model. For instance, H_i^k is the cost of recourse action type 1 and should be multiplied by the relevant failure type which is A_i^k to compute the extra travel cost type 1. Also, G_i^k should be multiplied by B_i^k and so on.

Finally, the total expected cost of recourse action in terms of distance can be computed as

$$E[\theta(R)] = \sum_{k=1}^{m} E[\theta(r_k)] \tag{8}$$

subject to:

$$\sum_{j\in V}\sum_{k\in K}\left(X_{ijk}+Y_{ijk}\right)=1,\quad \forall i\in V, \tag{9}$$

$$\sum_{j\in V}\left(X_{0jk}+Y_{0jk}\right)=1,\quad \forall k\in K, \tag{10}$$

$$\sum_{i\in V}\left(X_{i0k}+Y_{i0k}\right)=1,\quad \forall k\in K, \tag{11}$$

$$\sum_{i\in V}X_{ijk}-\sum_{i\in V}X_{jik}=0,\quad \forall j\in V,\ k\in K, \tag{12}$$

$$\sum_{i\in V}Y_{ijk}-\sum_{i\in V}Y_{jik}=0,\quad \forall j\in V,\ k\in K, \tag{13}$$

$$P\left\{\sum_{i\in v}\xi_i\sum_{j\in v}X_{ijk}<2\left(Q_k+Q_r\right)\right\}\cong 1 \tag{14}$$

If $v_i \in$ PVR or CVR

$$P\left\{\sum_{i\in v}\xi_i\sum_{j\in v}Y_{ijk}<2\left(Q_k\right)\right\}\cong 1 \quad \text{If } r_k\in \text{PTR or ST} \tag{15}$$

$$X_{ijk}\in\{0,1\},\quad Y_{ijk}\in\{0,1\},\quad \forall i,j\in V,\ k\in K, \tag{16}$$

where, r_k is the kth route and m is the number of routes including PTR, PVR and CVR. X_{ijk} is equal to 1 if customers i and j (edge $(i, j) \in E$) are serviced by a complete vehicle (the kth truck with a trailer), and 0 otherwise. Y_{ijk} is equal to 1 if edge $(i, j) \in E$ is used by the kth truck (without trailer). Equation (9) gives information about each customer that must be serviced by a single truck or complete vehicle. Equation (10) is concerned with starting each route from the depot and going to exactly one customer using vehicle k. Equation (11) is similar to (10); however, it shows the route terminates by leaving one customer. Equations (12) and (13) indicate the flow on the route to be followed by vehicle k. Equations (14) and (15) are the capacity constraints for the vehicle route and the truck route, respectively, and these ensure the feasibility of servicing customers, and q_i is the expected value of the stochastic customers' demands.

3. Metaheuristic Algorithms to Solve TTRPSD

Metaheuristics use two principal methods to improve the solution from that of heuristic in terms of different performance criteria. These methods are known as local search method [26] and population search method [35]. When using the local search method, one should know that a thorough search of the solution space is implemented by moving the existing solution to another likely solution in its neighborhood at each step. Tabu search (TS) and simulated annealing (SA) are the most well-known algorithms in this area [36]. The population search includes upholding a pool of good parent solutions and then reassociating them to create new offsprings. Genetic algorithm (GA) is the main example in this principle. To solve TTRPSD, multipoint simulated annealing (M-SA), tabu search and memetic algorithms (MA) have been used. MAs fit into the category of the evolutionary algorithms (EAs) where LS procedures are used to distribute the search area and enhance the search to refine the individual. In fact, MA is a hybrid algorithm which combines the population and LS procedures to improve the initial solutions.

In MAs, the population P consists of a set of individuals generated randomly. Each individual is called a "chromosome," which is simply a permutation of the set of n nodes (customers) $\{1, 2, \ldots, n\}$ and N dummy zeroes (artificial stores or the root of ST). This idea was initially proposed by Beasley [37] as part of a route-first cluster-second heuristic, and was then used by Prins [38]. The presentation of the solution for TTRPSD is explained more as follows. Initially, parents are selected based on tournament selection. Then, in each stage, offspring are randomly generated from the initial population. First, two chromosomes (parents) are randomly selected and two new chromosomes (offspring) are produced by crossover operation. The functional value is computed by the chromosomes. The new solutions are compared with the existing solutions. If a new solution is better than an existing solution, the algorithm replaces the existing solution with the new one. Then the new solution can be improved by mutations and LS procedures. This procedure is continued until the termination condition occurs.

In M-SA, firstly, the number of predetermined initial solutions should be produced and it is known as the best solutions. In each iteration, the algorithm produces some new solutions from the neighborhood of the current solutions and finds the best one between them and chooses this as a new solution. If a new solution appears to be better than the current solution, this new solution is termed as a best solution and the procedure is continued. In this process, the number of predetermined iterations is applied in each temperature level to improve the possibility of a set of appropriate solutions. However, sometimes the algorithm occurs in local optima. The procedure may escape from this issue by accepting worse solution with some conditions. This new solution with a worse objective function value can be accepted as the current solution with a small probability determined by the Boltzmann function, $\exp(-\Delta/KT)$, where K is a predetermined constant and T is the current temperature in Boltzmann function [18].

The presentation of the TS algorithm to solve TTRPSD is explained as follows. At each iteration, the algorithm explores the solution space and produces a new solution from the neighbourhood of the current solution. Even if, the objective function value becomes worse with this new solution. A tabu mechanism is put in place to prevent the process from cycling over a sequence of solutions. An intuitive way to prevent cycles is to forbid the process from returning to previously encountered solutions, which is achieved by exploiting excessive bookkeeping. Some attributes of the past solutions are registered and any solution possessing these attributes may

not be considered, and temporarily declared tabu for number of iterations (it is called tabu tenure).

3.1. Initial Solution. In this paper, this method is used to produce an initial solution considering the following assumptions. The initial solution can be used for M-SA, TS and MA. Consider

$$P\left\{\sum_{v_i \in r_k} \xi_i < 2\left(Q_k + Q_r\right)\right\} \cong 1, \quad \text{If } r_k \in \text{PVR, CVR,}$$
$$P\left\{\sum_{v_i \in r_k} \xi_i < 2\left(Q_k\right)\right\} \cong 1, \quad \text{If } r_k \in \text{PTR, ST.} \tag{17}$$

All customers are classified as TCs and VCs. A string of numbers represents an initial solution, which is denoted by the set $\{1, 2, \ldots, n\}$ and N dummy zeroes. The parameter N is utilized to terminate the ST or the route. This parameter is only used to predict the number of routes or STs. If demand was deterministic and failure could not occur, this parameter could be $\lfloor \sum_i q_i/(Q_k) \rfloor$, where q_i is customer's demand and Q_k is the truck capacity. However, since demand is stochastic and failure can occur a maximum of once, the parameter N can be computed by $\lfloor \sum_i q_i/(2Q_k) \rfloor$ where $\lfloor \cdot \rfloor$ denotes the largest integer, which is equal to or smaller than the enclosed number and q_i is the expected value of customers' demand. In the first $n + N$ positions, the ith nonzero number denotes the ith customer to be serviced. VC can be serviced either by a complete vehicle or a single truck and the service vehicle type can be of 1 or 0. Therefore, the vehicle type for VC is 1 if it is serviced by a single truck. Otherwise, its vehicle type is 0 and must be serviced by a complete vehicle. Hence, TCs must be serviced only by a single truck; therefore there is no need to mention it in the solution [18].

The presentation of the solution is explained more as follows. In this solution, the first number addresses the first customer that is to be served on the first route. A PTR is decided for the route, if a single truck is to service the first customer on a route. Without violating the following assumptions $P\{\sum_{v_i \in r_k} \xi_i < 2(Q_k + Q_r)\} \cong 1$ if $r_k \in$ PVR and CVR or $P\{\sum_{v_i \in r_k} \xi_i < 2(Q_k)\} \cong 1$ if $r_k \in$ PTR or ST to represent the servicing sequence, from left to right, one by one, other customers are added to the route. Note that if it is a complete vehicle on the CVR main tour or on a PVR, the vehicle capacity may be $(Q_k + Q_r)$ or on a CVR ST or on a PTR, if it is a single truck, it may be Q_k and it all depends on the vehicle type in use for the service. If, in the solution representation, the next customer to be served is zero, the vehicle will come back to the depot or the parking place. If it is on a CVR ST, the ST will be terminated, because the vehicle returns to the parking place. If not, it is on a CVR main tour, on a PVR or on a PTR. Consequently, the vehicle goes back to the depot and the route is terminated. It is worth mentioning that when recourse cost occurs, it should be considered and computed.

In the solution representation, generation of a new route will be occurred, if the termination of previous route has been taken place and there are still customers yet to be serviced. Therefore, the next customer will be selected for a new route. A TTRP solution is given by this solution representation, without violating the assumption and it can be verified. On the other hand, by using this solution representation, the number of the vehicles that are used might exceed the available vehicles. As a result, after the solution representation has generated the routes, in order to decrease the number of the vehicles that are used a procedure for route combination is considered necessary. The procedure of the route combination, checks the possibility of combining the two existing routes. However, this combination process must not violate the constraint of the capacity of the vehicle and if there are routes that can be combined together without the violation of this constraint they will be merged without any modification. This process goes on until the number of the used vehicles is not greater than the number of the vehicles available or it will stop if there are not any more routes that can be combined together without violating the assumptions. If the outcome solution still persists on using more vehicles than available, for each additional trailer or truck that is used, a cost of P, as a penalty, is added to the objective function in order to make the solutions of this type undesirable [9].

3.2. Simulated Annealing to Solve TTRPSD. Simulated annealing uses some parameters for improving the solutions, such as T_0, T_f, I_{iter}, K, N_{non}, α, P, n_{pop}, n_{move}, $I_{\text{iterpertemp}}$, where T indicates the thermodynamic temperature, which is used to simulate the system at different temperatures, whereas T gradually cools down from an initial high temperature (T_0) to a final low temperature (T_f). It means that the procedure will be stopped when the temperature reaches lower than T_f. I_{iter} represents the number of iterations during the procedure. K is the Boltzmann constant which is used in the probability function. N_{non} denotes the maximum number of allowable iterations in temperature. The algorithm will be terminated if the number of reductions exceed the N_{non} without improving the best cost. α is the coefficient for controlling the cooling scheme. P is the penalty cost associated with the number of extra vehicles used. Hence it is not allowed to use more than the number of available vehicles in 21 benchmark instance problems that are used for TTRPSD, the penalty cost is assigned to be very large (e.g., $P = 10^5$ is considered for this model) to prevent from this issue. n_{pop} is the number of initial solution that should be considered for producing new solutions. n_{move} is the number of move from current solution to generate new solutions. It means that each current solution can produce n_{move} new solutions and then the best one should be chosen as new solution. $I_{\text{iterpertemp}}$ is the number of iterations in each temperature T.

In M-SA, it is important to define an appropriate method for finding the effective neighbors to improve an existing solution. A random neighborhood structure including swap, reversion, insertion, and change of service vehicle type of VCs is used for generating an appropriate neighborhood from a current solution. In each iteration, the algorithm generates a new solution Y from the current solution X by using swap, reversion, insertion and change of service vehicle type randomly.

The swap is carried out by selecting two customers randomly and swapping them to generate a new solution from the current solution. The reversion is performed by selecting two numbers (customers) from the string of numbers representing the current solution then reversing the route from bigger number to smaller one. The insertion is performed by selecting two customers randomly and inserting the first customer immediately after the second one. The change of service vehicle type of VCs is fulfilled by selecting a vehicle customer randomly. If it is serviced by a single truck in the current solution, its service vehicle type will be changed to a complete vehicle and vice versa. It means that the vehicle service type of a selected VC will be changed from 1 to 0 or 0 to 1. For example, VC customer servicing in the main-tour by a complete vehicle will be serviced on a subtour by a single truck and vice versa.

The probability of selecting swap, reversion, insertion and change service vehicle type methods will be set to 0.25. For each of them as it is assumed that each of these methods has an equal chance to improve the solution. In addition, failure may occur. Therefore, the failure should be diagnosed in the route and recourse action cost should be calculated for each failure.

The M-SA procedure is as follows. At first, the current temperature is set to be the initial temperature and the algorithm generates the initial solutions X_i randomly. In addition, the best solution $X_{i\text{best}}$ and the best objective function considering recourse cost, if occurred, are set to be X_i and $\text{obj}(X_i, P)$, respectively. In each iteration, the algorithm generates new solutions Y_i from its neighborhood and evaluates its objective function value. Let Δ_i represents the differences between objective functional values from X_i and Y_i. Therefore, $\Delta_i = \text{obj}(Y_i, P) - \text{obj}(X_i, P)$. The probability of replacement of the current solution with the next solution is as follows

$$\begin{aligned} &1, \quad \Delta \le 0, \\ &\exp\left(\frac{-\Delta_i}{KT}\right), \quad \Delta > 0. \end{aligned} \tag{18}$$

If $\Delta_i \le 0$, it means that the next solution is better than the current solution and should be replaced for the current solution. If $\Delta_i > 0$, it means that the next solution is not appropriate. However, as it mentioned, since escaping from trapping in local search, these kinds of solutions will be accepted with $\exp(-\Delta_i/KT)$ probability by generating a random number $r \in [0, 1]$ and replacing the solution X_i with Y_i if $r < \exp(-\Delta_i/KT)$.

After accomplishing the number of iterations, the current temperature will be decreased according to the formula $T = \alpha T$, where $0 < \alpha < 1$. As there are more chances to find a better solution, the algorithm uses a local search procedure, which sequentially performs 2-Opt, swap, reversion, insertion and change of service vehicle types in every three temperature reductions [9].

If the current temperature T becomes lower than T_f or the number of reductions exceeds the N_{non} without improving the best cost, the algorithm will be terminated.

3.3. Tabu Search Algorithm to Solve TTRPSD. Similar to M-SA, TS also needs to start its procedure from an initial solution in the solution space. This initial solution can be infeasible. For this purpose, the initial solution method which is explained in previous section can be used for TS. TS algorithm tries to encourage the procedure to explore that part of the solution space which has not been visited yet. This purpose can be attained by preventing the reverse moves. The reversal of previous moves should become tabu for prohibiting a return to the previous solutions [39]. Cycle avoidance can be obtained over the tabu tenure. At each iteration, the inverse modification is identified as a tabu and occurred in the tabu list. In this paper, four operators (swap, reversion, insertion, and change of service vehicle type of VCs) have been considered for finding new solutions. In addition, tabu tenure η is generated randomly with integer uniform distribution from $[r, s]$. Four tabu lists TABU_i ($i = 1, 2, 3, 4$) are considered to store the relevant inverse modifications for four operators. If the tabu status $\text{TABU}_i(j, k)$ is less or equal 0, it means that $\text{arc}(j, k)$ is not tabu and vice versa.

Tabu moves can be overridden if the aspiration criterion is satisfied. If the tabu solution has less objective function value, it can be overridden.

The TS algorithm is explained as follows. At first, the relevant symbols are mentioned as follows:

(i) S: set of candidate solutions;

(ii) t: iteration counter;

(iii) t_{non}: current number of iteration without improvement;

(iv) t_n: maximum number of allowable iterations;

(v) $t_{\max}$: maximum number of iterations;

(vi) n_c: current number of generated candidate solutions;

(vii) $n_{\max}$: maximum number of candidate moves;

(viii) x^*: best-found solution for complete vehicle;

(ix) y^*: best-found solution for single truck;

(x) $T(x, y)$: the objective value;

(xi) $P(x, y)$: the penalized objective value.

Tabu Search Heuristic

(1) Generate an initial solution.

(2) Initialize the parameters: $A_1 = A_2 = 1$, $A_3 = A_4 = 1000$, $S = \emptyset$, $t = 0$, $n_c = 0$, $t_{\text{non}} = 0$, $\delta_i = 0$, $n_{\max}$, $t_{\max}$, t_n, r, s.

(3) Update the data: $x \to x^*$, $y \to y^*$, $T(x, y)$ and $P(x, y)$.

(4) If $n_c \ge n_{\max}$, go to (3).

(5) Generate random neighborhood $N(x, y)$ and moves from current solution to new solution.

(6) Update $t + 1 \to t$, $n_c + 1 \to n_c$, $S = S \cup \{x_{n_c}\} \cup \{y_{n_c}\}$.

(7) Evaluate all solutions in S by procedure 1 and set $i = 1$, $x_{\text{best}} = \emptyset$, $y_{\text{best}} = \emptyset$ and $P(x, y) = \infty$ and insert all nontabu solutions in $N'(x, y)$.

(8) If $i > n_c$, go to (10).

(9) If $(x_i, y_i) \in N'(x, y)$ and $P(x_i, y_i) < P(x_{\text{best}}, y_{\text{best}})$, $(x_{\text{best}}, y_{\text{best}}) = (x_i, y_i)$ and $P(x_{\text{best}}, y_{\text{best}}) = P(x_i, y_i)$; else if the solution is feasible, and $T(x_i, y_i) < T^*(x^*, y^*)$, some parameters should be updated as follows. $(x_{\text{best}}, y_{\text{best}}) = (x_i, y_i)$, $P(x_{\text{best}}, y_{\text{best}}) = P(x_i, y_i)$ and $i + 1 \rightarrow i$ and go to (8).

(10) If $P(x_{\text{best}}, y_{\text{best}}) \geq P^*(x, y)$, set $t_{\text{non}} + 1 \rightarrow t_{\text{non}}$, else update: $(x_{\text{best}}, y_{\text{best}}) \rightarrow (x, y)$, $(x_{\text{best}}, y_{\text{best}}) \rightarrow P^*(x, y)$.

(11) If $(x_{\text{best}}, y_{\text{best}})$ is feasible and $T(x_{\text{best}}, y_{\text{best}}) < T^*(x^*, y^*)$, then set $(x_{\text{best}}, y_{\text{best}}) \rightarrow (x^*, y^*)$. In addition, $T(x_{\text{best}}, y_{\text{best}}) \rightarrow T^*(x^*, y^*)$ and update the values of E_k, F_k, G_k, H_k and δ_i.

(12) Set $\text{TABU}_i(j,k) - 1 \rightarrow \text{TABU}_i(j,k)$. It should be mentioned that the corresponding inverse modifications are tabooed for a tabu tenure η, which is generated randomly from $[r, s]$.

(13) If $\text{mod}(t, 10) = 0$, then A_i should be updated. In addition, $\delta_i = 0$ and $t + 1 \rightarrow t$.

If $t \leq t_{\max}$ and $t_{\text{non}} \leq t_n$, then go to (5); else the algorithm is terminated and the best solution found is (x^*, y^*).

3.4. Memetic Algorithm to Solve TTRPSD. To employ the MA, various crossovers, mutations and local search approaches were applied. The order crossover (OX) and partial-mapped crossover (PMX) were considered as the representations of permutation. Goldberg and Holland (1988) were the first ones to propose the PMX [40]. The operator of the PMX is an extended version of the two-point crossover. The two-point crossover is applied for a binary string and by using this operator, the occurrence of some infeasible solutions might be expected. However, some procedures are used by PMX for repairing these infeasible solutions by fixing the solutions resulted from the two point crossover [41, 42]. First, a random selection of two positions in the chromosome is performed and then the subchromosomes, which have been located in between these positions, are replaced. After that, the relations of the two sections of mapping are decided. Finally, the rest of the customers are randomly arranged according to the discovered relations in the remaining positions [40]. For the TTRPSD solution, this operator of PMX has been considered. Figure 1 depicts a PMX sample representation. This figure indicates that two samples, which are considered as the chosen parent chromosomes and according to PMX, two new offspring are generated. At first, the subchromosomes (the shadow customers) are selected. After that, all of the customers in this subchromosome are substituted and at the end, the rest of the customers are arbitrarily allocated consistent with the developed relations.

The OX operator uses different methods for repairing the procedure. This operator has been successfully applied for the TSP and VRP [37]. First, a subchromosome from one parent is randomly selected. Then the new offspring is made by dropping the subchromosome into the same position. Finally, the remaining customers are arranged according to their positions in the other parent. A sample representation of an OX is illustrated in Figure 2.

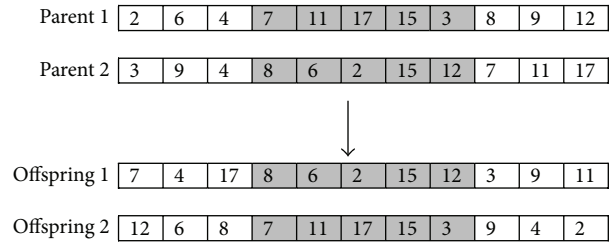


Figure 1: Partial-mapped crossover for TTRPSD.

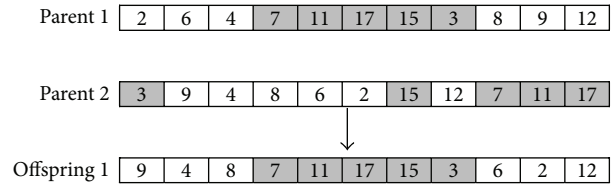


Figure 2: Order crossover for TTRPSD.

Various mutation types such as swaps, insertions, inversions, displacements and changes in service vehicle type for the VCs have been used by MA in the production of offspring. By random selection of subchromosomes and putting them in a random situation, the implementation of the displacement mutation is performed. Swap is implemented by random selection of two customers and changing them for creation of new solutions out of the current solution. Reversion is performed by selection of two chromosome customers and overturning the route from the greater number to the smaller number. The achievement of insertion is obtained by selecting two random customers and inserting the first customer exactly after the second one. The fulfilment of the change in service vehicle type for the VCs is achieved by choosing a random VC. If in the current solution, it is serviced by a single truck, then its complete vehicle type will be changed to a service vehicle type and vice versa. This means that the selected VC vehicle service type will be changed from 0 to 1 or 1 to 0. For instance, a VC that has been serviced by a complete vehicle in the main tour will be serviced by a single truck on a ST and vice versa. Different types of mutations applied in the MA are illustrated in Figures 3–6.

After applying the crossovers and mutation procedures, the approaches of LS are used for the improvement of the chromosomes in the candidates' pool. Three different procedures of LS, change ST root, two-point exchange (TPE), and one-point movement (OMP) are applied in the MA. Each one of the approaches can be selected randomly with probability of equal value.

A customer is moved in OPM from one particular route to one another. If this movement decreases the cost, then this movement can be accepted. In addition, only one customer is considered at a time. In movement execution, it is forbidden to move a TC customer to the main tour of the CVR or two

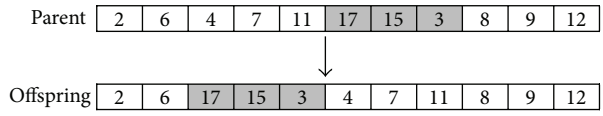


Figure 3: Displacement mutation for TTRPSD.

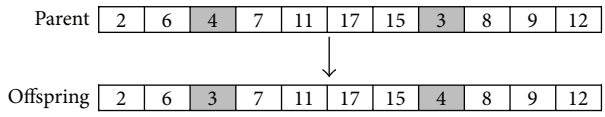


Figure 4: Swap mutation for TTRPSD.

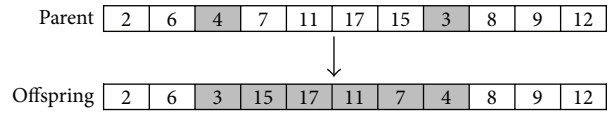


Figure 5: Reversion mutation for TTRPSD.

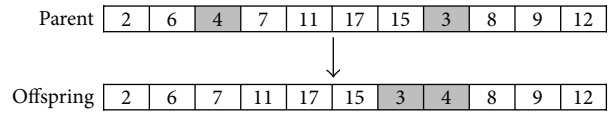


Figure 6: Insertion mutation for TTRPSD.

PVRs. At first, the customers on the main tour and the PTR are examined by the algorithm and then the examination of the ST customers is started.

Two customers of two different routes need to be replaced in a TPE. However, it is forbidden to move a TC customer into a PVR or into the main tour in the CVR. Considering the capacity of the truck, when customers are switched between two routes, all of the subroute continues to be feasible. In the aforementioned procedures, the position of the root node is never changed. This is due to an improvement in the solution when a number of root nodes are substituted. In this case, customer resequencing or root reselection is considered.

4. Computational Study

In this section, a real case study from an Iranian dairy company is provided. This study has been carried out with the aid of Pegah Co., a large dairy distribution company in Iran, whose products are distributed to more than 50,000 retailers (customers) in Iran and some other countries. Iran Dairy Industries Co. (IDIC) is the largest dairy producer in Iran with "PEGAH" brand. This factory produces some dairy products such as Pasteurized and UHT milk, flavored milk, pasteurized and UHT cream, a variety of cheese (process, slice, pizza, UF), different kinds of yoghurt, probiotic products (such as yoghurt, cheese, ice cream), and drinking yoghurt.

To implement the TTRPSD model for this case, 100 customers were selected based on their stochastic demands and the types of customers. To select the customers, the last 20 demands of each customer were realized and the customers of stochastic demands with the Poisson distribution were selected for this research. Then the customers' locations were determined to compute the travel distance matrix between the customers and the depot. Furthermore, the type of each customer was distinguished and the truck customers (TCs) and the vehicle customers (VCs) were classified into their respective groups, where 30 customers were TC and the remaining customers were VC. The demands were measured based on their weights. The company considered 5 trucks and 3 trailers to serve these customers. The capacity of a truck and a trailer are 2000 and 3000 kilograms, respectively.

4.1. Computational Results. The MA, M-SA and TS were coded by MATLAB 7.9.0 using a computer with a 2.4 GHz dual processor and 4 G RAM. This type of problem has not yet been considered and there is no scope to compare this solution with an existing one. To overcome this issue, the special 21 TTRPSD benchmark instances are modified which are derived from Chao [7] for the TTRP with deterministic demands. First, the benchmarks were solved in order to increase the validity of the aforesaid algorithms and to show the consistency of the results. Then the case study problem was solved using MA, M-SA and TS. Further, the case study problem was checked by sensitivity analysis to confirm the results.

To implement the benchmarks, the coordinates of vertices are kept the same as in the Chao's instances [7]. As said earlier, the customer demands are stochastic and are generated with the Poisson distribution having a mean value equal to the customer demand. However, due to increasing possibility of failure on the route, the capacity of a truck and trailer is less than Chao's benchmark, because the failure has rarely occurred with its capacities. Therefore, they are reduced considering $P\{\xi_i \le Q_k\} \cong 1$ for each customer.

Parameters used in the model may affect the quality of computational results. These parameters may be used for MA, M-SA or TS. In order to obtain better solutions, different values have been tested in initial experiments. These are: $P = 10000$; $\alpha = 0.99, 0.975, 0.95$; $I_{\text{iter}} = 20000, 40000, 60000, 80000, 100000$; $I_{\text{iterpertemp}} = 2, 4, 6, 8, 10$; $K = 1, 0.9, 0.8, 0.7, 0.6, 0.5$; $n_{\text{pop}} = 1, 3, 5, 7, 9$; $n_{\text{move}} = 1, 3, 5, 7, 9$; $T_0 = 1000$, $T_F = 0.1$ or $T_0 = 10$, $T_F = 0.001$; $N_{\text{non}} = 100, 200$; $[r, s] = [5, 10]$; $n_{\max} = 150$; $t_{\max} = 1000$; $t_n = 50, 100$; $P_c = 0.7, 0.75, 0.8$; $P_m = 0.2, 0.3, 0.4$; $P_l = 0.3, 0.4, 0.5$; $n = 50, 100$, where P_c, P_m, P_l and n are the probability of crossover, mutation, LS improvement and population size, respectively. Since it is not possible to use more than the predetermined number of vehicles, the penalty cost is considered too high. The parameters have been examined with different values and set to be $\alpha = 0.95$, $I_{\text{iter}} = 60000$, $K = 0.7$, $T_0 = 10$, $T_F = 0.001$, $n_{\text{pop}} = 5$, $n_{\text{move}} = 9$, $I_{\text{iterpertemp}} = 6$ and $N_{\text{non}} = 200$, $t_n = 50$, $P_c = 0.7$, $P_m = 0.3$, $P_l = 0.5$ and $n = 100$. They seem to give the best results and will be used for further computation.

Table 1 illustrates the benchmark results. Each set has been run 10 times and the best solutions of MA, M-SA and TS from the 10 runs are shown.

Table 1: The best solutions of TTRPSD benchmarks.

Problem ID	Number of VC	Number of TC	Truck number	Truck capacity	Trailer number	Trailer capacity	MA results	M-SA results	TS results
1	38	12	5	60	3	60	635.45	639.38	643.30
2	25	25	5	60	3	60	671.84	659.77	666.56
3	13	37	5	60	3	60	677.44	683.76	686.42
4	57	18	9	60	5	60	929.11	931.63	942.32
5	38	37	9	60	5	60	921.01	919.10	924.55
6	19	56	9	60	5	60	971.90	991.28	1007.85
7	75	25	8	100	4	60	934.34	931.83	926.79
8	50	50	8	100	4	60	879.80	898.86	911.33
9	25	75	8	100	4	60	956.87	949.57	954.27
10	113	37	12	100	6	60	1117.32	1126.66	1133.38
11	75	75	12	100	6	60	1220.02	1198.94	1228.06
12	38	112	12	100	6	60	1221.29	1230.32	1243.02
13	150	49	17	100	9	60	1312.69	1309.84	1305.68
14	100	99	17	100	9	60	1374.32	1377.01	1381.58
15	50	149	17	100	9	60	1473.65	1475.00	1482.67
16	90	30	7	100	4	60	1092.37	1129.42	1113.86
17	60	60	7	100	4	60	1139.52	1114.28	1118.90
18	30	90	7	100	4	60	1118.93	1127.65	1120.05
19	75	25	10	100	5	60	843.46	843.46	852.14
20	50	50	10	100	5	60	853.72	867.77	876.15
21	25	75	10	100	5	60	867.62	965.24	948.41
Avg.							1010.13	1017.65	1022.25

Table 1 indicates that the average results generated by the MA are better than M-SA and TS. However, the differences between these results are insignificant and the results show that these algorithms are effective for solving related problems.

Table 2 indicates that the average results obtained by the proposed MA were improved about 1.27 and 0.78 percent comparing with tabu search and multipoint simulated annealing, respectively. Consequently, the performance of the proposed MA is found slightly better than SA and TS. As the differences between these results are insignificant, the results obtained by MA, M-SA and TS can be accepted as the new solutions. Therefore, the results indicate that the algorithms are efficient and effective in solving TTRP with stochastic demands.

For every parameter, the algorithms were tuned sequentially, leaving the remaining parameters unchanged. For this problem, in total about 400 runs were undertaken during the sensitivity analysis (including running a parameter setting 10 times on the problem for each algorithm).

The results in kilometers which are obtained by MA, M-SA, and TS are presented in Tables 3–5, respectively. Each set has been run 10 times and the best (best-sol), the worst (worst-sol) and the average solutions (avg.-sol) from the 10 runs are shown. Also, the time taken for the best solutions is presented in the last column. As this type of problem was not solved earlier, the results cannot be compared with any data or earlier heuristic solutions. However, the problems were checked by sensitivity analysis. Now, the results can be compared.

Tables 2 to 4 show the best solutions obtained by MA, M-SA and TS are 205.484, 207.230, and 209.015, respectively. In addition, the comparison between the obtained results and the sensitivity analysis results confirm a very slight difference. Therefore, the applied MA, M-SA, and TS are efficient with confidence consistency of a reasonable time.

To indicate the convergence of the proposed approach, trends are shown in Figure 7. This study has presented the relation between the number of iteration and the obtained objective function value. As it can be noted from the figure, the improvement rate of the solution reduces as the number of the iteration increases and after a particular number of iteration, the achieved solution converges. Therefore, the quality of the solution may not be enhanced by a greater number of iteration.

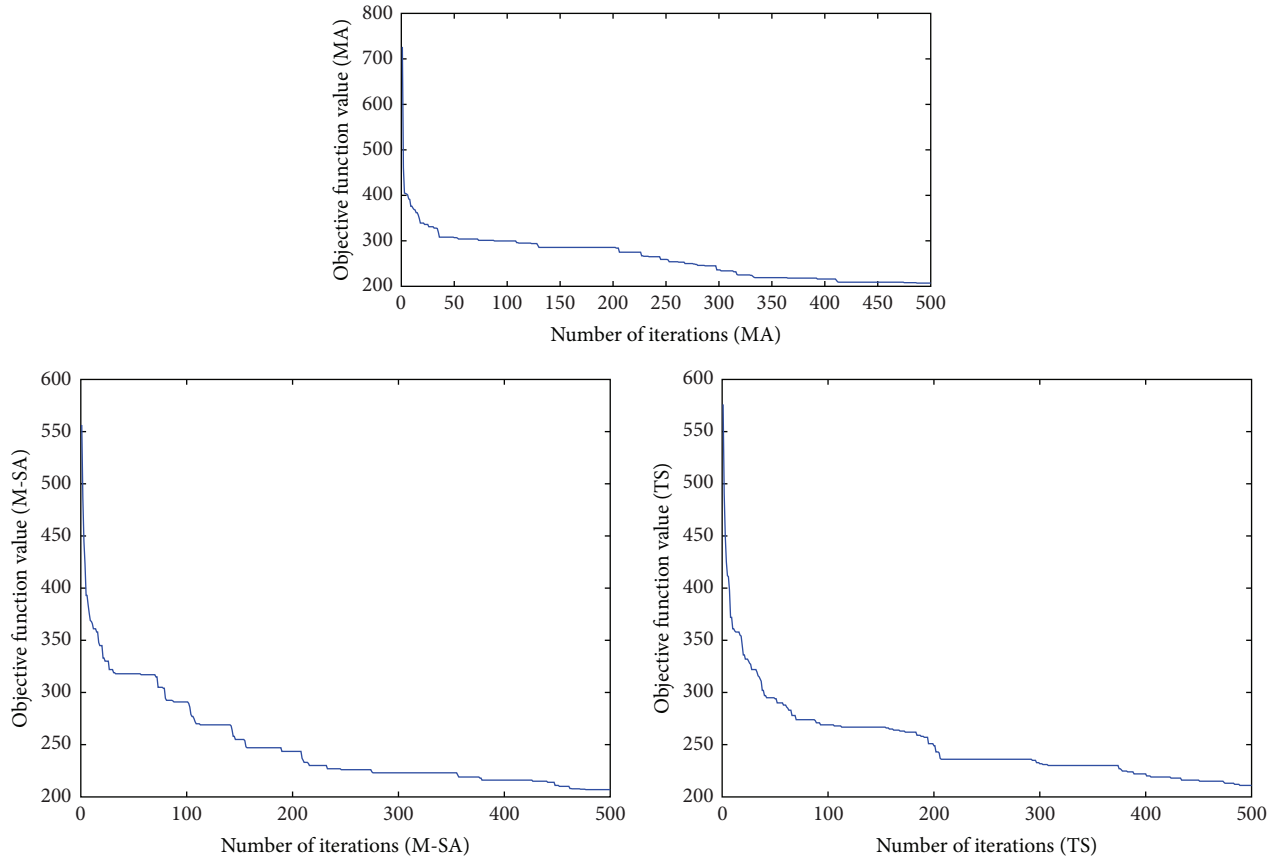


Figure 7: Convergence trend for the algorithms' solution.

In brief, the analyses show that all of the methods of TS, M-SA, and MA are greatly efficient and their efficiency is almost equal; however, it seems that MA indicates slightly more efficiency than the others.

5. Conclusion and Further Research Directions

This research introduced, formulated, and solved the real-world stochastic TTRP problem. Indeed, in case of most real-life problems, demands are varying and TTRP should be expanded for meeting stochastic demands in the name of TTRP with stochastic demands (TTRPSD). Most researchers, however, solved TTRP with deterministic demands.

This study modelled the TTRPSD problem within the framework of a stochastic program with recourse action (SPR). Three metaheuristic algorithms, namely, MA, M-SA, and TS have been applied to solve the problem. Firstly, twenty-one TTRP benchmarked problems have been modified for this model. Then MA, M-SA, and TS algorithms have been applied to solve these benchmarks to show the validity, consistency, and efficiency of the algorithms. The results are obtained in a reasonable time. As the differences between the results are not so much and the achieved solution converges after certain iterations, the aforesaid algorithms are efficient with confidence consistency.

Each set of three case studies problem ran for 10 times and the best, worst, and average results were compared. As the differences between the best, worst, and corresponding average solutions are insignificant, the algorithm is found capable of producing TTRPSD solutions consistently and the results are useful.

Furthermore, the problems have been tested by sensitive parameter analysis to realize the impact of the parameters. Since the results between the best, worst, and corresponding average solutions and the results obtained by sensitive parameter analysis are found closer; the results approved and indicated that the MA, M-SA, and TS are efficient and effective in solving this case study problem.

Future research may attempt to extend the TTRPSD by introducing other practical or real-world conditions, such as multiple time windows, time dependent travel times and multidepots. Also, work on TTRP with stochastic travel times to solve real-life problems in manufacturing and transportation systems can be considered. Consequently, new TTRP benchmark instance problems need to be modified for this purpose.

Conflict of Interests

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

Table 2: Compared the results obtained from proposed MA with TS and M-SA.

Problem ID	Comparison of best solution between proposed MA and TS results (MA-TS)/TS (%)	Comparison of best solution between proposed MA and M-SA results (MA-SA)/SA (%)
1	−1.22	−0.61
2	0.79	1.83
3	−1.31	−0.93
4	−1.40	−0.27
5	−0.39	0.21
6	−3.60	−1.96
7	0.81	0.27
8	−3.58	−2.12
9	0.27	0.76
10	−1.44	−0.84
11	−0.66	1.73
12	−1.78	−0.74
13	0.53	0.22
14	−0.53	−0.20
15	−0.61	−0.09
16	−1.93	−3.28
17	1.81	2.21
18	−0.10	−0.78
19	−1.02	0
20	−2.56	−1.62
21	−8.52	−10.11
Average	−1.27	−0.78

Table 3: The MA results and sensitivity.

Best-sol	Worst-sol	Avg-sol	Sensitivity analysis	Time (second)
205.484	214.498	208.389	204.673	647

Table 4: The M-SA results.

Best-sol	Worst-sol	Avg-sol	Sensitivity analysis	Time (second)
207.230	213.827	209.027	206.004	810

Table 5: The TS results.

Best-sol	Worst-sol	Avg-sol	Sensitivity analysis	Time (second)
209.015	218.984	211.582	208.716	759

Acknowledgment

This paper acknowledges the financial support from the University of Malaya Research Grant, UMRG Project RG139/12AET for this research.

References

[1] K. Lee, H. Cho, and M. Jung, "Simultaneous control of vehicle routing and inventory for dynamic inbound supply chain," *Computers in Industry*, vol. 65, no. 6, pp. 1001–1008, 2014.

[2] S. F. Ghannadpour, S. Noori, and R. Tavakkoli-Moghaddam, "Multiobjective dynamic vehicle routing problem with fuzzy travel times and customers' satisfaction in supply chain management," *IEEE Transactions on Engineering Management*, vol. 60, no. 4, pp. 777–790, 2013.

[3] C.-J. Liao, Y. Lin, and S. C. Shih, "Vehicle routing with cross-docking in the supply chain," *Expert Systems with Applications*, vol. 37, no. 10, pp. 6868–6873, 2010.

[4] J. Zhang, W. Wang, Y. Zhao, and C. Cattani, "Multiobjective quantum evolutionary algorithm for the vehicle routing problem with customer satisfaction," *Mathematical Problems in Engineering*, vol. 2012, Article ID 879614, 19 pages, 2012.

[5] G. Laporte, "The vehicle routing problem: an overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 3, pp. 345–358, 1992.

[6] G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet, "Classical and modern heuristics for the vehicle routing problem," *International Transactions in Operational Research*, vol. 7, pp. 285–300, 2000.

[7] I.-M. Chao, "A tabu search method for the truck and trailer routing problem," *Computers & Operations Research*, vol. 29, no. 1, pp. 33–51, 2002.

[8] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, pp. 80–91, 1959.

[9] S.-W. Lin, V. F. Yu, and S.-Y. Chou, "Solving the truck and trailer routing problem based on a simulated annealing heuristic," *Computers and Operations Research*, vol. 36, no. 5, pp. 1683–1692, 2009.

[10] U. Derigs, M. Pullmann, and U. Vogel, "Truck and trailer routing—problems, heuristics and computational experience," *Computers & Operations Research*, vol. 40, no. 2, pp. 536–546, 2013.

[11] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "A matheuristic for the truck and trailer routing problem," *European Journal of Operational Research*, vol. 230, no. 2, pp. 231–244, 2013.

[12] J. C. Gerdessen, "Vehicle routing problem with trailers," *European Journal of Operational Research*, vol. 93, no. 1, pp. 135–147, 1996.

[13] R. Cordone and R. W. Calvo, "A heuristic for the vehicle routing problem with time windows," *Journal of Heuristics*, vol. 7, no. 2, pp. 107–129, 2001.

[14] B. M. Baker and M. A. Ayechew, "A genetic algorithm for the vehicle routing problem," *Computers and Operations Research*, vol. 30, no. 5, pp. 787–800, 2003.

[15] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: a taxonomic review," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1472–1483, 2009.

[16] S. Scheuerer, "A tabu search heuristic for the truck and trailer routing problem," *Computers & Operations Research*, vol. 33, no. 4, pp. 894–909, 2006.

[17] M. L. Fisher and R. Jaikumar, "A generalized assignment heuristic for vehicle routing," *Networks*, vol. 11, no. 2, pp. 109–124, 1981.

[18] S.-W. Lin, V. F. Yu, and C.-C. Lu, "A simulated annealing heuristic for the truck and trailer routing problem with time windows," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15244–15252, 2011.

[19] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 780–794, 2010.

[20] J. G. Villegas, C. Prins, C. Prodhon, A. L. Medaglia, and N. Velasco, "A GRASP with evolutionary path relinking for the truck and trailer routing problem," *Computers and Operations Research*, vol. 38, no. 9, pp. 1319–1334, 2011.

[21] Y. Kao, M.-H. Chen, and Y.-T. Huang, "A hybrid algorithm based on ACO and PSO for capacitated vehicle routing problems," *Mathematical Problems in Engineering*, vol. 2012, Article ID 726564, 17 pages, 2012.

[22] F. A. Tillman, "The multiple terminal delivery problem with probabilistic demands," *Transportation Science*, vol. 3, no. 3, pp. 192–204, 1969.

[23] D. J. Bertsimas, "A vehicle routing problem with stochastic demand," *Operations Research*, vol. 40, no. 3, pp. 574–585, 1992.

[24] M. Gendreau, G. Laporte, and R. Séguin, "A tabu search heuristic for the vehicle routing problem with stochastic demands and customers," *Operations Research*, vol. 44, no. 3, pp. 469–477, 1996.

[25] G. Laporte, F. V. Louveaux, and L. van Hamme, "An integer *L*-shaped algorithm for the capacitated vehicle routing problem with stochastic demands," *Operations Research*, vol. 50, no. 3, pp. 415–423, 2002.

[26] H. Lei, G. Laporte, and B. Guo, "The capacitated vehicle routing problem with stochastic demands and time windows," *Computers & Operations Research*, vol. 38, no. 12, pp. 1775–1783, 2011.

[27] X. Li, P. Tian, and S. C. H. Leung, "Vehicle routing problems with time windows and stochastic travel and service times: models and algorithm," *International Journal of Production Economics*, vol. 125, no. 1, pp. 137–145, 2010.

[28] W. R. Stewart Jr. and B. L. Golden, "Stochastic vehicle routing: a comprehensive approach," *European Journal of Operational Research*, vol. 14, no. 4, pp. 371–385, 1983.

[29] G. Laporte, F. Louveaux, and H. Mercure, "Models and exact solutions for a class of stochastic location-routing problems," *European Journal of Operational Research*, vol. 39, no. 1, pp. 71–78, 1989.

[30] M. Gendreau, G. Laporte, and R. Séguin, "Stochastic vehicle routing," *European Journal of Operational Research*, vol. 88, no. 1, pp. 3–12, 1996.

[31] K. C. Tan, C. Y. Cheong, and C. K. Goh, "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation," *European Journal of Operational Research*, vol. 177, no. 2, pp. 813–839, 2007.

[32] J. E. Mendoza, B. Castanier, C. Guéret, A. L. Medaglia, and N. Velasco, "A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands," *Computers and Operations Research*, vol. 37, no. 11, pp. 1886–1898, 2010.

[33] H. N. Psaraftis, "Dynamic vehicle routing: status and prospects," *Annals of Operations Research*, vol. 61, no. 1, pp. 143–164, 1995.

[34] N. Secomandi and F. Margot, "Reoptimization approaches for the vehicle-routing problem with stochastic demands," *Operations Research*, vol. 57, no. 1, pp. 214–230, 2009.

[35] B. Xie and S. An, "Applying genetic algorithm to vehicle routing problem with stochastic travel times," *Dynamics of Continuous Discrete and Impulsive Systems. Series A*, vol. 13, pp. 693–697, 2006.

[36] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the Operational Research Society*, vol. 57, no. 10, pp. 1143–1160, 2006.

[37] J. E. Beasley, "Route first—cluster second methods for vehicle routing," *Omega*, vol. 11, no. 4, pp. 403–408, 1983.

[38] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 12, pp. 1985–2002, 2004.

[39] X. Zhang, S. Zhong, Y. Liu, and X. Wang, "A framing link based Tabu search algorithm for large-scale multi-depot vehicle routing problems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 152494, 13 pages, 2014.

[40] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine Learning*, vol. 3, no. 2-3, pp. 95–99, 1988.

[41] R. Tavakkoli-Moghaddam, A. R. Saremi, and M. S. Ziaee, "A memetic algorithm for a vehicle routing problem with backhauls," *Applied Mathematics and Computation*, vol. 181, no. 2, pp. 1049–1060, 2006.

[42] C. Prins, "Two memetic algorithms for heterogeneous fleet vehicle routing problems," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 6, pp. 916–928, 2009.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 562462, 9 pages
http://dx.doi.org/10.1155/2015/562462

Research Article

Energy-Saving Generation Dispatch Using Minimum Cost Flow

Zhan'an Zhang and Xingguo Cai

School of Electrical Engineering and Automation, Harbin Institute of Technology, Heilongjiang 150001, China

Correspondence should be addressed to Zhan'an Zhang; zhangzhananhit@163.com

Received 16 July 2014; Revised 31 August 2014; Accepted 31 August 2014

Academic Editor: Dongdong Ge

This study uses a minimum cost flow method to solve a dispatch problem in order to minimize the consumption of coal in the dispatching of a thermal power system. Low-carbon generation dispatching is also considered here since the scheduling results are consistent with energy-saving generation dispatch. Additionally, this study employs minimum coal consumption as an objective function in considering the output constraints, load balance constraints, line loss, ramp rate limits, spinning reverse needs, prohibited operating zone requirements, security constraints, and other common constraints. The minimum cost flow problem, considering the loss of network flow, is known as a generalized network flow problem, which can be expressed as a quadratic programming problem in mathematics. Accordingly, the objective function was solved by LINGO11, which was used to calculate a network in a single time; a continuous period dispatch plan was obtained by accumulating each period network flow together. This analysis proves it feasible to solve a minimal cost flow problem with LINGO11. Theoretical analysis and numerical results prove the correctness and effectiveness of the proposed method.

1. Introduction

There seems to be rather compelling evidence that global warming is an issue that we seriously need to be concerned about today [1, 2]. Carbon dioxide accounts for 80% of the greenhouse effect, and rising carbon dioxide levels are the main cause of global warming [3, 4]. China pledged to reduce its carbon intensity by 40–45% by 2020 based on 2005 levels. By the end of 2012, 71.5% of China's generation capacity was from thermal power, of which 92.5% was the product of coal-fired generation [5]. Coal-fired electricity consumes about 50% of China's coal production, and the CO_2 emissions from power generation account for 40% of the total CO_2 emissions in China. Accordingly, China's power industry has implemented energy-saving generation dispatching (ESGD) and low-carbon generation dispatching (LCGD).

ESGD is one of the most important problems in power system operations requiring load demand at minimum total fuel cost while accounting for various unit and system constraints. The ESGD model is an optimization problem that considers linear and nonlinear characteristics, including power balance constraints, generation limit constraints, node voltage constraints, ramp rate limits, spinning reverse needs, prohibited operating zone requirements, and security constraints, among others [6]. In this paper, low-carbon generation dispatching is also considered, since the scheduling results are consistent with energy-saving generation dispatching. In practical scheduling applications, a daily scheduling period is generally divided into 24 or more intervals; therefore, dispatching of each period can be solved as a static optimization problem.

Many mathematical techniques have been developed and applied to dispatch problem such as linear programming [7], interior-point method [8], Lagrangian relaxation algorithm [9], quadratic programming [10] and other traditional algorithms. These algorithms essentially need some problem simplification such that the problem is linear or convex. Thus, a true global minimum cannot be guaranteed [11]. The dynamic programming method [12] has also been successfully used in solving the dispatch problems; however, this method may result in "curse of dimensionality." More recently, the metaheuristic algorithms such as particle swarm optimization (PSO) [13], genetic algorithm (GA) [14], simulated annealing (SA) [15], evolutionary programming (EP) [16], and ant colony optimization (ACO) [17], have also been considered in the context of economic dispatch. However, the biggest

problem that metaheuristic algorithms faced is that the computational efficiency is rather low. The related parameters are not easy to set up and the computational time is long. It is not easy to implement when applied to large electrical power systems. Their highly heuristic nature usually leads to suboptimal solutions. Differential evolution (DE) is a stochastic search based method [18], which can present a simple structure, fast convergence speed, and robustness. However, DE fast convergence might lead the direction of the search toward a local optimal and premature solution. Essentially, the economic dispatch problem is a large scale nonlinear programming problem. In pursuit of the optimal solution for economic dispatch, various hybrid methods have been investigated and implemented [5, 19–22]. Unfortunately, these hybrid algorithms normally take lengthy calculation time when compared with the mathematical optimization methods. Moreover, previous algorithms rarely considered network structures, we use a network flow method here to make up for the deficiency, because the network flow method can well retain the topology of the network.

Networks provide a useful way for modeling power system problems and are used extensively in power system dispatching [23, 24]. As an important network problem, ESGD problems can be formulated and solved as minimum cost flow problems when the cost is a quadratic function of the power, which is a nonlinear, minimum cost flow problem.

The main objective of this study is to introduce convex quadratic programming to solve the ESGD problem, since coal consumption and network losses are all convex functions of the power flow through a network. In order to do that this study employed LINGO11 to solve quadratic programming problems accounting for linear and nonlinear equality constraints and inequality constraints. Accordingly, a minimum cost flow algorithm was used to solve the ESGD model and calculate a network within a single moment. From that process, a continuous period of ESGD planning was obtained by accumulating period network flow results. This process confirmed that the minimal cost flow method was successful for solving the ESGD problem and, therefore, has value for these types of applications.

2. Problem Formulation

2.1. The Mathematical Model of ESGD Problem. The ESGD problem determines the optimal schedule of the available generating units to simultaneously minimize the generation cost rate and meet the load demand of a power system while meeting various equality and inequality constraints. This mathematical model can be described as follows:

$$\min \sum_{t=1}^{T} \sum_{i=1}^{N} F_{it}\left(P_{it}\right) = \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\alpha_i P_{it}^2 + \beta_i P_{it} + \gamma_i\right), \quad (1)$$

where F_{it} is the cost function of the ith generator; α_i, β_i, and γ_i are the cost coefficients of the ith generator; P_{it} is the power of the ith generator at t time; and N represents the number of generators committed to the operating system.

2.2. The Mathematical Model of Low-Carbon Generation Dispatching. Achieving the lowest carbon emission is the target of low-carbon generation dispatching. This mathematical model can be described as follows:

$$\min \sum_{t=1}^{T} \sum_{i=1}^{N} d_{it}\left(P_{it}\right), \quad (2)$$

where $d_{it}(P_{it})$ is the electrical-carbon characteristic function. This formula represents the CO_2 emissions when the output of unit i is P_{it} at t time (t/h), which can be expressed as

$$d_t = \frac{e}{q\eta} P_t = 2.77038 F\left(P_t\right), \quad (3)$$

where e is the CO_2 emission coefficient of the fuel used in a power source; the standard coal emission factor is 2.77, which means that 2.77 kg CO_2 can be discharged for every 1 kg of standard coal burnt; q is the calorific value of unit fuel, which is 8.14 kWh/kg of standard coal; and η is power generation efficiency, which can be expressed as

$$\eta = \frac{3600 P_t}{29308 F\left(P_t\right) \times 10^3} \times 100\%, \quad (4)$$

where 3600 is the electric heating value (kJ/kWh) and 29308 is the calorific value of standard coal (kJ/kg).

If the coal consumption function and electrical-carbon characteristic satisfies the following relationship

$$\frac{F_1\left(P_{1t}\right)}{d_1\left(P_{1t}\right)} = \cdots = \frac{F_i\left(P_{1t}\right)}{d_i\left(P_{1t}\right)} = \lambda, \quad d_i\left(P_{1t}\right) \neq 0, \qquad F_i\left(P_{1t}\right) = 0, \quad d_i\left(P_{1t}\right) = 0, \quad (5)$$

where λ is a constant, then we can get

$$\min \sum_{t=1}^{T} \sum_{i=1}^{N} F_{it}\left(P_{it}\right) = \lambda \min \sum_{t=1}^{T} \sum_{i=1}^{N} d_{it}\left(P_{it}\right); \quad (6)$$

then the 2 kinds of scheduling results are consistent [25].

2.3. Constraints. The objective function needs to satisfy the following constraints.

(1) Power Constraints. The power constraints include generator output, transformer capacity, and line transmission limits. Consider

$$P_{i\min} \leq P_{i,t} \leq P_{i\max} \quad i = 1, 2, \ldots, N, \quad (7)$$

where $P_{i\min}$ is the lower and $P_{i\max}$ is the upper output limit of unit i, respectively.

(2) Node Voltage Constraints. Consider

$$U_{i\min} \leq U_{i,t} \leq U_{i\max} \quad i = 1, 2, \ldots, N, \quad (8)$$

where $U_{i,t}$ is the voltage of node i at t time and $U_{i\min}$ is the lower and $U_{i\max}$ is the upper limit of node i, respectively.

(3) Power Balance Constraints. Consider

$$\sum_{i=1}^{N} P_{i,t} = P_{Dt} + P_{Lt} \quad i = 1, 2, \ldots, N, \quad (9)$$

where P_D is the total load demand and P_L is the transmission network losses, which is a function of unit power outputs that can be represented using the B coefficients:

$$P_L = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} P_i B_{0i} + B_{00}, \tag{10}$$

where B_{ij} is the loss coefficient square matrix; B_{0i} is the loss coefficient vector; and B_{00} is the loss coefficient constant [26].

(4) Operation Ramp Rate Limits. The power output of a practical generator cannot be adjusted instantaneously without limits. The operating range for all online units is restricted by their ramp rate limits during each dispatch period. Therefore, the dispatch output of a generator should be limited by the constraints of up and down ramp rates [27], which are given as follows:

$$\begin{aligned} P_{i,t-1} - P_{i,t} &\le DR_i \quad \text{if generation decreases} \\ P_{i,t} - P_{i,t-1} &\le UR_i \quad \text{if generation increases,} \end{aligned} \tag{11}$$

where DR_i and UR_i are the ramp-down and ramp-up rate limits of the ith thermal unit, respectively [28]. If the unit ramp rate limits are considered, the real power operating limits are modified as follows:

$$\max\left(P_i^{\min}, P_{i,t-1} - DR_i\right) \le P_{i,t} \le \min\left(P_i^{\max}, P_{i,t-1} + UR_i\right). \tag{12}$$

(5) Spinning Reserve. The added spinning reserve factor must be considered to prevent a sudden large load to the system or a failure in a certain large unit requirement. This condition can be explained as follows:

$$\sum_{i=1}^{N} \left(P_{i\max} \times U_{it}\right) \ge \left(P_{Dt} + R_t\right), \tag{13}$$

where R_t is the spinning reserve in the tth hour and U_{it} is the ON and OFF status of the ith conventional unit at the t period ($U_{it} = 0$ represents OFF status and $U_{it} = 1$ represents ON status) [6].

(6) Prohibited Operating Zone. The prohibited operating zone is the range of prohibited output power resulting from the physical limitations of machine components, steam valves, vibration in the shaft bearing, and other conditions that can cause discontinuity in the electrical energy cost curve. Therefore, some units must be considered as prohibited zones in practical operation. The feasible operating zones of thermal units can be described as follows [29]:

$$\begin{aligned} P_i^{\min} &\le P_i \le P_{i,1}^l, \\ P_{i,j-1}^u &\le P_i \le P_{i,j}^l, \\ P_{i,z_i}^u &\le P_i \le P_i^{\max}, \\ j &= 2, 3, \ldots, z_i, \end{aligned} \tag{14}$$

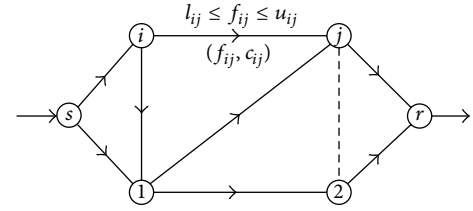


Figure 1: A network flow diagram.

where z_i is the number of prohibited zones in the ith generator curve; j is the index of prohibited zone of the ith generator; and $P_{i,j}^u$ and $P_{i,j}^l$ are the upper and lower limits of the jth prohibited zone of unit i, respectively.

(7) Security Constraints. For secure operation, the transmission line loading (S_i) is restricted by its upper limit as follows:

$$S_i \le S_i^{\max} \quad (i = 1, 2, \ldots, NL), \tag{15}$$

where NL is the total number of lines [30].

3. The Minimal Cost Flow Method

3.1. Network Flow Theory Introduction. Network flow problems can be described with graph theory, where a graph represents a network of nodes and connecting arcs. If a path exists between any two pairs of vertices in a graph, then that graph is a connected graph where each arc has a specific direction.

As depicted in Figure 1, the common abstraction that models a flow network is a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of edges over these vertices.

Arc(i, j) is the edge connecting nodes i and j, which has a predetermined direction, flow f_{ij}, the cost of the unit flow c_{ij}, and flow rate limit (as shown in Figure 1). Each edge (i, j) has a flow f_{ij} that defines the number of units of the commodity that flows from i to j. An edge also has a capacity limit that constrains the number of units that can flow over that edge. Here, c_{ij} denotes the cost per unit flow transported directly from vertex i to vertex j [31]. When no units are flowing over an edge, then the arc is outlined in a dotted line.

A special source vertex $s \in V$ produces units of a commodity that flow through the edges of the graph to be consumed by a sink vertex $r \in V$ known as the receiving node. The rest nodes, for which there are both outflow and inflow arcs, are termed intermediate points.

The following criteria must be satisfied for any feasible flow through a network.

(1) Capacity Constraint. The flow f_{ij} through an edge cannot exceed the capacity of the edge u_{ij}, $l_{ij} \le f_{ij} \le u_{ij}$. If an edge (i, j) does not exist in the network, then $u_{ij} = 0$.

(2) Flow Conservation. Aside from the source vertex s and sink vertex r, each vertex $i \in V$ must satisfy the property for which the sum of f_{ij} for all edges (i, j) in E (the flow into i) must equal the sum of f_{ij} for all edges $(i, j) \in E$ (the flow out of i). This property ensures that flow is neither produced nor consumed in the network, except at s and r.

(3) Skew Symmetry. For consistency, the quantity f_{ij} represents the net flow from vertex i to j. This means that it must be the case that $f_{ij} = -f_{ji}$, which holds, even if both edges (i, j) and (j, i) exist in a directed graph (Figure 1) [32].

3.2. The Minimal Cost Flow Method. The minimum cost flow problem determines the minimum total cost under the specified flow while taking into account the arc cost. The objective function can be expressed as follows:

$$\min \sum_{(i,j)\in E} c_{ij} f_{ij}. \tag{16}$$

If c_{ij} is a constant, the minimum cost flow problem can be solved efficiently since it can be formulated as a linear programming problem.

Both the coal consumption and the network losses are convex functions of power flow within a power network. In minimal cost flow problems, the cost of the arc is a convex function of the flow on the arc and that is a convex cost flow problem. When c_{ij} is a linear function of flow f_{ij}, then the objective function changes into a quadratic programming problem. Not only the objective function but also the constraints include quadratic terms of f_{ij}. Accordingly, the cost function of unit flow can be expressed as

$$\frac{df(P_i)}{dP} = 2\alpha_i P_i + \beta_i, \tag{17}$$

where $f(P_i)$ stands for $F(P_i)$ or P_L. Hence, the objective function model can be expressed as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} (\alpha_i f_i + \beta_i) f = \min \sum_{i=1}^{n} \left(\alpha_i f_i^2 + \beta_i f_i\right) \\ \text{s.t.} \quad & \sum_{(i,j)\in E} f_{ij} - \mu_{ji} \sum_{(j,i)\in E} f_{ji} = b(i), \quad \forall i \in G. \end{aligned} \tag{18}$$

In this model, the value of $b(i)$ depends on the nature of node i, where $b(i) > 0$ if node i is a supply node, $b(i) < 0$ if node i is a demand node, and $b(i) = 0$ when node i is a transshipment node [33].

In a general minimum cost flow problem, the arc is the conservation of the flow, and the flow entering an arc equals the flow leaving the arc. This assumption is reasonable in many practical application scenarios; however, the power flow will diminish when flowing through the grid due to resistance, which, in this study, is considered an issue associated with generalized flow problems. In generalized flow problems, arcs might consume or generate flow. If f_{ij} units of flow enter an arc(i, j), then $\mu_{ij} f_{ij}$ units arrive at node j, where μ_{ij} is a positive multiplier associated with the arc. If $0 \le \mu_{ij} \le 1$, the arc is lossy, and, if $1 \le \mu_{ij} \le \infty$, the arc is gainy. The problem becomes a general minimal cost flow problem when $\mu_{ji} = 1$ and $b(i) = 0$. In this study, all the arcs are loss arcs due to the resistance, which means that $(0 \le \mu_{ij} \le 1)$ [34].

There is no single standard algorithm that can always be used to solve convex programming problems. There have been many algorithms for solving convex quadratic programming problems, such as the Lemke method, the interior-point method [35, 36], the effective set method [37, 38], and an ellipsoid algorithm, for example, each having its own advantages and disadvantages.

In this study, LINGO11, very mature software widely used in various mathematical optimization problems, is used to solve the quadratic programming algorithm problems.

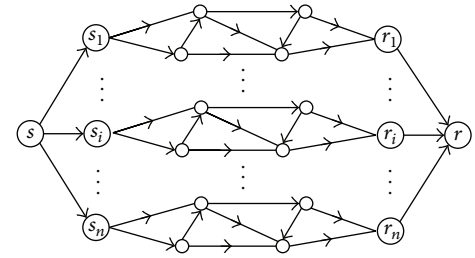


Figure 2: Multiperiod network flow model.

3.3. Multiperiod Network Flow Model. A continuous time network flow is constituted by combining each network flow chart for a single period of time. A multiperiod model is a three-dimensional model with each layer corresponding to a single-period model (as shown in Figure 2). Each s_i is united to form new s, and, in the same way, to form a new r. Arc(s, s_i) represents the output and arc(r_i, r) represents the load. Considering each period together, a continuous-time power scheduling plan can be obtained. It is merely a schematic diagram, and the practical applications in power system may be more complex than that in Figure 2. For instance, the direction of the arcs in each layer may vary with different operating conditions. In order to simplify the analysis, we assume that the lines and transformers meet their capacity constraints.

The above structure model (Figure 2) represents a typical generalized network flow programming problem. It can be expressed as

$$\min C^T X, \tag{19}$$

satisfied with

$$AX = b, \tag{20}$$

where C represents the m-dimensional arc cost vector; X represents the m-dimensional arc flow; b represents an n-dimensional vector injected nodes; and A represents $n \times m$ dimensional node-arc incidence matrix [39].

4. The Solution Process Diagram

The steps in the procedure of ESGD (the low-carbon dispatching calculation process is the same) can be described as follows (as shown in Figure 3).

Step 1. Determine the load P_t at t time according to the load forecasting curve, where T represents the total of time periods.

Step 2. Put P_t as the flow at sink vertex.

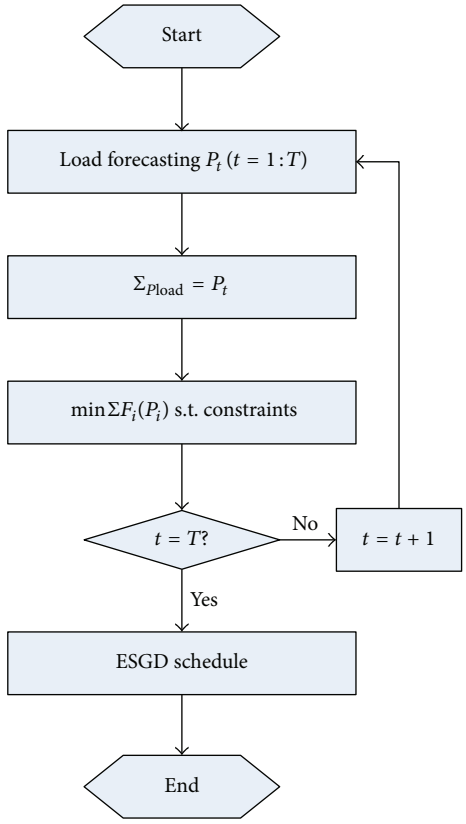


Figure 3: Flowchart of solution procedure.

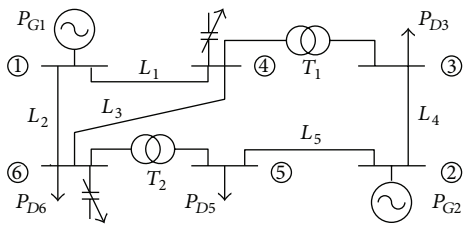


Figure 4: IEEE 6-bus power system diagram.

Step 3. Solve the solution according to (1), satisfying the constraints (see (7) to (15)).

Step 4. If $t = T$, then output ESGD scheduling results; otherwise, if $t < T$, then let $t = t + 1$ and go to Step 2.

5. Example Analysis

For this study, we calculated a modified IEEE 6-bus power system, which contains 2 generators, 2 transformers and 5 transmission lines (as shown in Figure 4).

The parameters in the modified system are shown in Tables 1, 2, and 3. In the tables, r_0 is the resistivity of a conductor per unit length, L is the length of lines, R and R_T are the resistance of lines and transformers, respectively, ΔP_k represents short circuit loss of a transformer, and α_i, β_i, and γ_i refer to the cost coefficients of the ith generator (in Table 3).

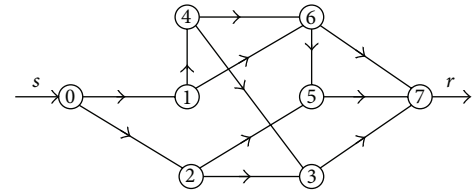


Figure 5: Network diagram.

Table 1: The parameters of transmission lines.

Line	V (kV)	Model	r_0 (Ω/km)	L (km)	R (Ω)
L_1	110	LGJ-185	0.17	71.2	12.1
L_2	110	LGJ-185	0.17	71.2	12.1
L_3	110	LGJ-185	0.17	71.2	12.1
L_4	10	LGJ-120	0.242	10	2.42
L_5	10	LGJ-120	0.242	10	2.42

Table 2: The parameters of transformers.

T	V (kV)	Model	ΔP_k (kw)	R_T (Ω)
T_1	110/11	SFPL1-50000/110	250	1.21
T_2	110/11	SFPL1-50000/110	250	1.21

Table 3: The parameters of generators.

G	Capacity (MW)	Output limit	α_i	β_i	γ_i
G_1	100	$50 \le P_{G_1} \le 100$	0.0014	0.25	2.5
G_2	100	$50 \le P_{G_2} \le 100$	0.0018	0.18	5.0

The corresponding cost flow network diagram is shown in Figure 5, in which the directed diagram of the system includes 8 vertexes and 12 arcs.

In order to consider network losses, the net loss unit is turned into a standard coal consumption unit; that is, 1 MW = 0.123 t/h. In this model, the equivalent calorific value of 1 kg standard coal is 7000 kcal. The total-cost optimization model can be expressed as follows:

$$\begin{aligned}
\min \quad & F(x) = \left(0.25x_{01} + 0.0028x_{01}^2\right) \\
& + \left(0.18x_{02} + 0.0036x_{02}^2\right) \\
& + 0.0002 \times 0.123 \times \left(x_{43}^2 + x_{65}^2\right) \\
& + 0.0004 \times 0.123 \\
& \times \left(x_{25}^2 + x_{23}^2\right) + 0.002 \times 0.123 \\
& \times \left(x_{14}^2 + x_{46}^2 + x_{16}^2\right) \\
\text{s.t.} \quad & x_{01} - x_{14} - x_{16} = 0 \\
& x_{02} - x_{25} - x_{23} = 0 \\
& x_{23} - 0.0002x_{23}^2 + x_{43} - 0.0001x_{43}^2 - x_{37} = 0 \\
& x_{14} - 0.001x_{14}^2 - x_{46} - x_{43} = 0
\end{aligned}$$

$$
\begin{aligned}
&x_{65} - 0.0001x_{65}^2 + x_{25} - 0.0002x_{25}^2 - x_{57} = 0\\
&x_{46} - 0.001x_{46}^2 + x_{16} - 0.001x_{16}^2 - x_{65} - x_{67} = 0\\
&x_{37} + x_{57} + x_{67} = P_{D3} + P_{D5} + P_{D6}\\
&50 \le x_{01} \le 100, \quad 50 \le x_{02} \le 100,\\
&0 \le x_{14} \le 100, \quad 0 \le x_{46} \le 100,\\
&0 \le x_{16} \le 100, \quad 0 \le x_{43} \le 50,\\
&0 \le x_{25} \le 100, \quad 0 \le x_{65} \le 50,\\
&0 \le x_{23} \le 100, \quad 0 \le x_{67} \le 100,\\
&0 \le x_{57} \le 100, \quad 0 \le x_{37} \le 100,
\end{aligned}
\tag{21}
$$

where x refers to the flow in the network; then the objective function is obviously a convex cost flow problem and a quadratic programming method can be used to solve it. When the load changes, as shown in Figure 8, the differing distribution of the minimum cost flow can be concluded and, therefore, a continuous dispatch result can be obtained.

In order to simplify the analysis, we assumed that all units run within a safe operation area. The ramp rate limit of 100 MW unit is 2 MW/min, so the ramp rate limits can meet the requirements created when the time interval is an hour.

We also assumed that the reactive power can be compensated locally and the system does not transmit reactive power. The voltage of nodes was kept within their specified ratings, and the reserve capacity of the grid was assumed to meet operational requirements. Additionally, the test was performed on an Intel (R) Pentium (R) CPU 2.13 GHz 2.0 GB RAM with LINGO11, and the average iteration times were 58. The maximum flow that can pass the network and the corresponding minimal cost distribution is shown in Figure 6. The minimum flow distribution of the minimum cost flow is shown in Figure 7.

The load and corresponding minimal coal consumption are listed in Table 4 and shown in Figure 8, respectively. Where T is the time period (h), P is the power of load (MW) and F is the coal consumption (t/h).

Electricity and minimal coal consumption accumulation are listed in Table 5 and shown in Figure 10, respectively. S refers to the accumulation of coal consumption (t) and W refers to accumulation of electricity (MWh).

It can be seen that the accumulated electricity will gradually increase with the increase of time. The same trend occurs with the accumulated minimal coal consumption. Unit scheduling results are shown in Figure 9.

According to the electrical-carbon conversion relations, 2.77 tons of CO_2 can be emitted for 1 ton of standard coal consumed. In addition, the minimum CO_2 emissions and the accumulation curve can also be obtained (as shown in Figures 11 and 12).

The value of CO_2 emissions and its accumulation are listed in Table 6, where C_e and C_a refer to CO_2 emissions and its accumulation value, respectively.

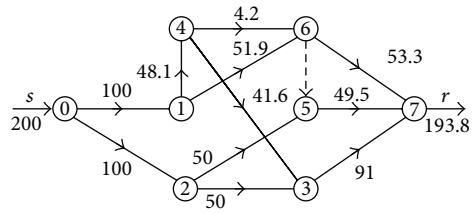


Figure 6: The minimal cost flow under maximum flow.

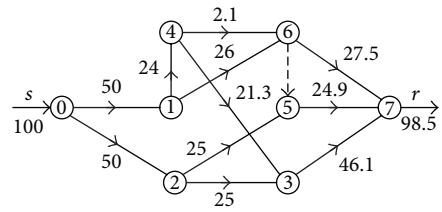


Figure 7: The minimal cost flow distribution under minimum flow.

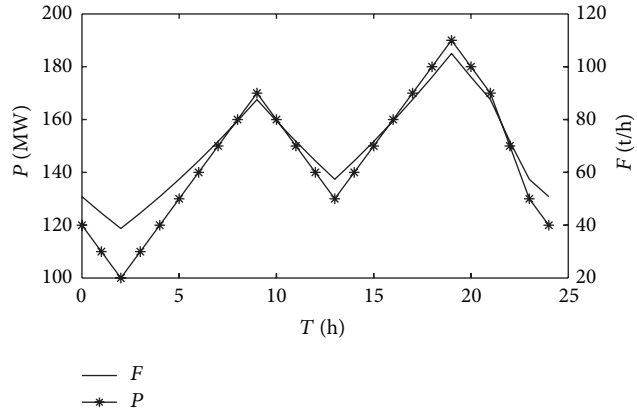


Figure 8: Load and coal consumption curve.

We can see that the emissions of CO_2 change with the power load by the same regularity, which reflects that the objectives of low-carbon generation dispatching and energy-saving generation dispatching are consistent.

In order to support our findings, we also solved this same problem by invoking the CPLEX solver in the MATLAB toolbox YALMIP [40]. The extended solver of LINGO11 includes Branch-and-Bound solver, Global solver, and Multistart solver, while the solver in CPLEX is mainly based on interior-point method. If these two different methods come to a same conclusion, then the result concluded is credible. The result by CPLEX produced the same solution garnered using LINGO11, which proves the correctness of the method used in this study.

6. Conclusions

The minimal cost flow algorithm can be used to dispatch the power system and can reflect the system network topology, making it fairly easy to consider system constraints.

Table 4: Load and coal consumption.

T (h)	1	2	3	4	5	6	7	8	9	10	11	12
P (MW)	110	100	110	120	130	140	150	160	170	160	150	140
F (t/h)	44.58	38.74	44.58	50.80	57.39	64.36	71.71	79.45	87.58	79.45	71.71	64.36
T (h)	13	14	15	16	17	18	19	20	21	22	23	24
P (MW)	130	140	150	160	170	180	190	180	170	150	130	120
F (t/h)	57.39	64.36	71.71	79.45	87.58	96.10	105.0	96.10	87.58	71.71	57.39	50.80

Table 5: Electricity and coal consumption accumulation.

T (h)	1	2	3	4	5	6	7	8	9	10	11	12
S (t)	44.6	83.3	127.9	178.7	236.1	300.5	372.2	451.6	539.2	618.6	690.4	754.7
W (MWh)	110	210	320	440	570	710	860	1020	1190	1350	1500	1640
T (h)	13	14	15	16	17	18	19	20	21	22	23	24
S (t)	812.1	876.5	948.2	1027	1115	1211	1316	1412	1500	1571	1629	1680
W (MWh)	1770	1910	2060	2220	2390	2570	2760	2940	3110	3260	3390	3510

Table 6: The value of CO_2 emissions and CO_2 emissions accumulation.

T (h)	1	2	3	4	5	6	7	8	9	10	11	12
C_e (t)	123.5	107.3	123.5	140.7	159.0	178.3	198.6	220.1	242.6	220.1	198.6	178.3
C_a (t)	123.5	230.8	354.3	495.0	654.0	832.3	1031	1251	1494	1714	1912	2091
T (h)	13	14	15	16	17	18	19	20	21	22	23	24
C_e (t)	159.0	178.3	198.6	220.1	242.6	266.2	290.9	266.2	242.6	198.6	159.0	140.7
C_a (t)	2249	2428	2626	2846	3089	3355	3646	3912	4155	4353	4513	4653

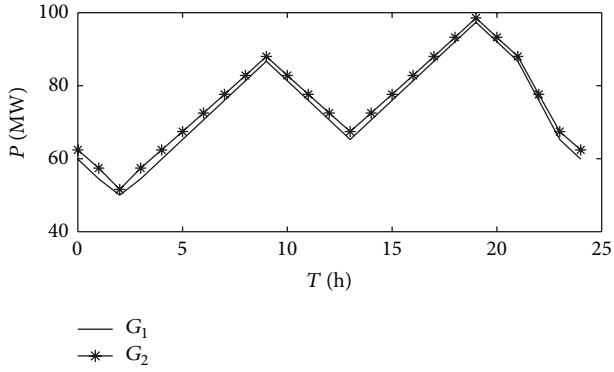


Figure 9: Outputs of generators.

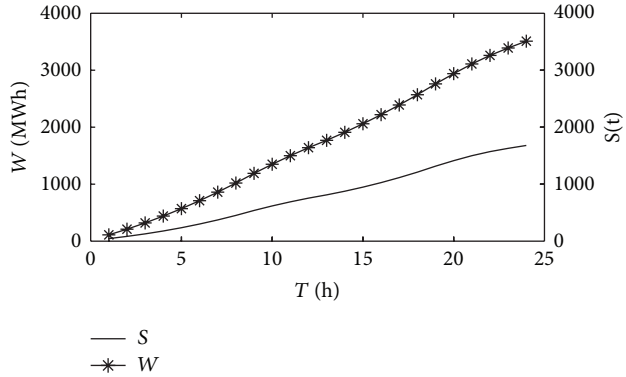


Figure 10: Electricity and coal consumption accumulation curve.

The method is simple, rapid, and clear in terms basic concepts. The example analysis proves that the method is feasible and practical.

In order to simplify the analysis, this study only considers the energy-saving scheduling of thermal units, without considering the effects of hydropower, wind power, and other renewable energy sources: although they do not consume coal and discharge CO_2, they do have an impact on load-flow distribution and network losses in power systems. Accordingly, the next step will be to initiate research to study the influences of renewable energy sources on ESGD in power systems.

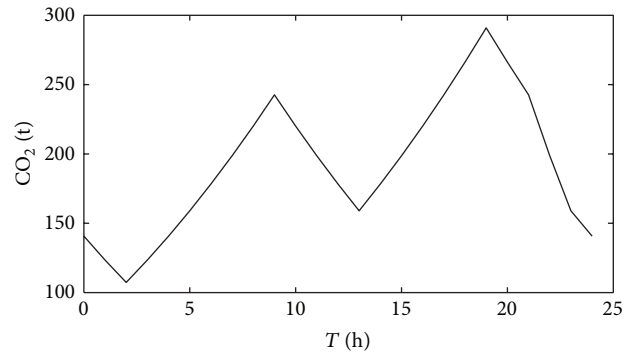


Figure 11: CO_2 emissions curve.

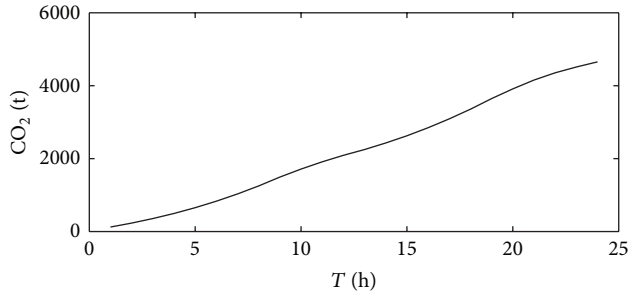


Figure 12: CO_2 emissions accumulation curve.

Conflict of Interests

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

Acknowledgments

The authors would like to acknowledge the guidance and support from their colleagues and supervisor, along with great thanks to Prof. Galen Leonhardy for editorial assistance.

References

[1] W. Yu and B. G. Xin, "Governance mechanism for global greenhouse gas emissions: a stochastic differential game approach," *Mathematical Problems in Engineering*, vol. 2013, Article ID 312585, 13 pages, 2013.

[2] H. M. Ndiritu, K. Kibicho, and B. B. Gathitu, "Influence of flow parameters on capture of carbon dioxide gas by a wet scrubber," *Journal of Power Technologies*, vol. 93, no. 1, pp. 9–15, 2013.

[3] Y.-C. Chang, T.-S. Chan, and W.-S. Lee, "Economic dispatch of chiller plant by gradient method for saving energy," *Applied Energy*, vol. 87, no. 4, pp. 1096–1101, 2010.

[4] J. Kotowicz and P. Lukowicz, "Influence of chosen parameters on economic effectiveness of a supercritical combined heat and power plant," *Journal of Power Technologies*, vol. 93, no. 5, pp. 323–329, 2013.

[5] C.-T. Cheng, S.-S. Li, and G. Li, "A hybrid method of incorporating extended priority list into equal incremental principle for energy-saving generation dispatch of thermal power systems," *Energy*, vol. 64, no. 1, pp. 688–696, 2014.

[6] G.-C. Liao, "A novel evolutionary algorithm for dynamic economic dispatch with energy saving and emission reduction in power system integrated wind power," *Energy*, vol. 36, no. 2, pp. 1018–1029, 2011.

[7] R. A. Jabr, A. H. Coonick, and B. J. Cory, "A homogeneous linear programming algorithm for the security constrained economic dispatch problem," *IEEE Transactions on Power Systems*, vol. 15, no. 3, pp. 930–936, 2000.

[8] R. A. Jabr, A. H. Coonick, and B. J. Cory, "A primal-dual interior point method for optimal power flow dispatching," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 654–662, 2002.

[9] W. Ongsakul and N. Petcharaks, "Unit commitment by enhanced adaptive Lagrangian relaxation," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 620–628, 2004.

[10] H. Zhong, Q. Xia, Y. Wang, and C. Kang, "Dynamic economic dispatch considering transmission losses using quadratically constrained quadratic program method," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2232–2241, 2013.

[11] N. Sinsuphan, U. Leeton, and T. Kulworawanichpong, "Optimal power flow solution using improved harmony search method," *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2364–2374, 2013.

[12] J. J. Hargreaves and B. F. Hobbs, "Commitment and dispatch with uncertain wind generation by dynamic programming," *IEEE Transactions on Sustainable Energy*, vol. 3, no. 4, pp. 724–734, 2012.

[13] J. Soares, M. Silva, T. Sousa, Z. Vale, and H. Morais, "Distributed energy resource short-term scheduling using Signaled Particle Swarm Optimization," *Energy*, vol. 42, no. 1, pp. 466–476, 2012.

[14] C. Yaşar and S. Özyön, "Solution to scalarized environmental economic power dispatch problem by using genetic algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 38, no. 1, pp. 54–62, 2012.

[15] M. H. Gomes and J. T. Saraiva, "A market based active/reactive dispatch including transformer taps and reactor and capacitor banks using simulated Annealing," *Electric Power Systems Research*, vol. 79, no. 6, pp. 959–972, 2009.

[16] P. Somasundaram and K. Kuppusamy, "Application of evolutionary programming to security constrained economic dispatch," *International Journal of Electrical Power and Energy Systems*, vol. 27, no. 5-6, pp. 343–351, 2005.

[17] S. Pothiya, I. Ngamroo, and W. Kongprawechnon, "Ant colony optimisation for economic dispatch problem with non-smooth cost functions," *International Journal of Electrical Power and Energy Systems*, vol. 32, no. 5, pp. 478–487, 2010.

[18] M. Sharma, M. Pandit, and L. Srivastava, "Reserve constrained multi-area economic dispatch employing differential evolution with time-varying mutation," *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 3, pp. 753–766, 2011.

[19] J. S. Alsumait, J. K. Sykulski, and A. K. Al-Othman, "A hybrid GA-PS-SQP method to solve power system valve-point economic dispatch problems," *Applied Energy*, vol. 87, no. 5, pp. 1773–1781, 2010.

[20] J. Cai, Q. Li, L. Li, H. Peng, and Y. Yang, "A hybrid CPSO-SQP method for economic dispatch considering the valve-point effects," *Energy Conversion and Management*, vol. 53, no. 1, pp. 175–181, 2012.

[21] C. C. A. Rajan, "A solution to the economic dispatch using EP based SA algorithm on large scale power system," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 6, pp. 583–591, 2010.

[22] M. Basu, "Hybridization of bee colony optimization and sequential quadratic programming for dynamic economic dispatch," *International Journal of Electrical Power and Energy Systems*, vol. 44, no. 1, pp. 591–596, 2013.

[23] A. R. L. Oliveira, S. Soares, and L. Nepomuceno, "Optimal active power dispatch combining network flow and interior point approaches," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1235–1240, 2003.

[24] J. Zhu and J. A. Momoh, "Multi-area power systems economic dispatch using nonlinear convex network flow programming," *Electric Power Systems Research*, vol. 59, no. 1, pp. 13–20, 2001.

[25] C. Li, Y. Liu, Y. Cao, Y. Tan, C. Xue, and S. Tang, "Consistency evaluation of low-carbon generation dispatching and energy-saving generation dispatching," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 31, no. 31, pp. 94–101, 2011.

[26] S. S. Reddy, B. K. Panigrahi, R. Kundu, R. Mukherjee, and S. Debchoudhury, "Energy and spinning reserve scheduling for a wind-thermal power system using CMA-ES with mean learning technique," *International Journal of Electrical Power and Energy Systems*, vol. 53, no. 1, pp. 113–122, 2013.

[27] C.-C. Kuo, "Generation dispatch under large penetration of wind energy considering emission and economy," *Energy Conversion and Management*, vol. 51, no. 1, pp. 89–97, 2010.

[28] A. Immanuel Selvakumar, "Enhanced cross-entropy method for dynamic economic dispatch with valve-point effects," *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 3, pp. 783–790, 2011.

[29] R. Azizipanah-Abarghooee, T. Niknam, A. Roosta, A. R. Malekpour, and M. Zare, "Probabilistic multiobjective wind-thermal economic emission dispatch based on point estimated method," *Energy*, vol. 37, no. 1, pp. 322–335, 2012.

[30] M. A. Abido, "Multiobjective particle swarm optimization for environmental/economic dispatch problem," *Electric Power Systems Research*, vol. 79, no. 7, pp. 1105–1113, 2009.

[31] F.-R. Xie and R.-A. Jia, "Nonlinear fixed charge transportation problem by minimum cost flow-based genetic algorithm," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 763–778, 2012.

[32] G. T. Heineman, G. Pollice, and S. Selkow, *Algorithms in a Nutshell*, O'Reilly Media, Sebastopol, Calif, USA, 2008.

[33] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, McGraw-Hill, 9th edition, 2010.

[34] R. K. Ahuja, T. L. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, NJ, USA, 1993.

[35] F. Curtis and J. Nocedal, "Steplength selection in interior-point methods for quadratic programming," *Applied Mathematics Letters*, vol. 20, no. 5, pp. 516–523, 2007.

[36] M. D' Apuzzo and M. Marino, "Parallel computational issues of an interior point method for solving large bound-constrained quadratic programming problems," *Parallel Computing: Theory and Applications*, vol. 29, no. 4, pp. 467–483, 2003.

[37] W. Cheng, Z. Chen, and D.-H. Li, "An active set truncated Newton method for large-scale bound constrained optimization," *Computers & Mathematics with Applications*, vol. 67, no. 5, pp. 1016–1023, 2014.

[38] M.-T. Yu, T.-Y. Lin, and C. Hung, "Active-set sequential quadratic programming method with compact neighbourhood algorithm for the multi-polygon mass production cutting-stock problem with rotatable polygons," *International Journal of Production Economics*, vol. 121, no. 1, pp. 148–161, 2009.

[39] Z.-A. Zhang and X.-G. Cai, "Power purchase plan using minimal cost flow," *Computer Modelling and New Technologies*, vol. 18, no. 2, pp. 281–285, 2014.

[40] J. Löfberg, "YALMIP: a toolbox for modeling and optimization in MATLAB," in *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, pp. 284–289, Taipei, Taiwan, September 2004.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 637809, 10 pages
http://dx.doi.org/10.1155/2015/637809

Research Article

A Variable Depth Search Algorithm for Binary Constraint Satisfaction Problems

N. Bouhmala

Department of Technology and Maritime Innovation, Buskerud and Vestfold University College, P.O. Box 4, 3199 Borre, Norway

Correspondence should be addressed to N. Bouhmala; noureddine.bouhmala@hbv.no

Received 7 October 2014; Revised 5 March 2015; Accepted 1 April 2015

Academic Editor: Jianming Shi

The constraint satisfaction problem (CSP) is a popular used paradigm to model a wide spectrum of optimization problems in artificial intelligence. This paper presents a fast metaheuristic for solving binary constraint satisfaction problems. The method can be classified as a variable depth search metaheuristic combining a greedy local search using a self-adaptive weighting strategy on the constraint weights. Several metaheuristics have been developed in the past using various penalty weight mechanisms on the constraints. What distinguishes the proposed metaheuristic from those developed in the past is the update of k variables during each iteration when moving from one assignment of values to another. The benchmark is based on hard random constraint satisfaction problems enjoying several features that make them of a great theoretical and practical interest. The results show that the proposed metaheuristic is capable of solving hard unsolved problems that still remain a challenge for both complete and incomplete methods. In addition, the proposed metaheuristic is remarkably faster than all existing solvers when tested on previously solved instances. Finally, its distinctive feature contrary to other metaheuristics is the absence of parameter tuning making it highly suitable in practical scenarios.

1. Introduction

Organizations like companies or public institutions are confronted in their daily life with a large number of combinatorial optimization problems which occur in many different application domains such as Operations Research (e.g., scheduling and assignment), hardware design (verification and testing, placement and layout), financial decision making (option trading or portfolio management), or even biology (DNA sequencing). The domain of combinatorial optimization refers to optimization problems where the search space (i.e., the set of all feasible solutions) is discrete. The constraint satisfaction problem (CSP) which can model a wide spectrum of combinatorial optimization problems rising in the field of artificial intelligence has become an important field of study in both theoretical and applied computer science. Constraint technology is making a considerable commercial impact worldwide due to its ability to solve highly complex applications operating in the most difficult environment counting on first-class technology to perform the job. ILOG and Cosytec are two of the leading companies producing software based on this technology. A large number of systems based on the constraints technology have been developed. Examples include the APACHE system [1] used at Roissy Airport in Paris, PLAN system [2] which is a medium-long term scheduling system for aircraft assembly line scheduling, the COBRA system [3] that generates work plans for train drivers and conductors of North Western Trains in the UK, and TAP-AI which is a planning system for crew assignment in the airline SAS [4]. Disasters which have long impacted world nations, resulting in mass casualties and huge financial tolls where technology and humans have to work together hand-in-hand without fault, with every single step of a mission meticulously planned out, are another research area where solutions based on constraint technology have received a great attention lately [5, 6]. The handbook of Constraint Programming [7] lists example applications of several areas modeled as CSPs. The paper is organized as follows. Section 2 explains the constraint satisfaction problem. Section 3 provides a survey of methods used to solve the constraint satisfaction problem. Section 4 introduces the metaheuristic in detail. Section 5 presents the results while Section 6 concludes the paper.

2. CSP

The CSP consists of assigning values to variables while satisfying certain constraints. Constraints can be given explicitly, by listing all possible tuples or implicitly, by describing a relation in some mathematical form. As a domain example, consider problems that occur in production scheduling. Scheduling is concerned with the allocation of resources to activities with the goal of optimizing some performance objectives while satisfying certain restrictions or constraints. Depending on the problem posed, resources may refer to machines, humans, and so forth, activities could be manufacturing operations, objectives could be the minimization of the schedule length, and finally constraints may state the precedence relationship among activities as they govern the schedule solution.

A CSP is a tuple $\langle X, D, C\rangle$, where

(i) X is a finite set of variables: $X = \{X_1, X_2, \ldots, X_n\}$,

(ii) D is a finite set of domains: $D = \{D_{X_1}, D_{X_2}, \ldots, D_{X_n}\}$. Thus each variable $X_i \in X$ has a corresponding discrete domain D_{X_i} from which it can be instantiated,

(iii) $C = \{C_1, C_2, \ldots, C_k\}$ is a finite set of constraints. Each k-ary constraint restricts a k-tuple of variables $(X_1, X_2, \ldots, X_k)$ and specifies a subset of $D_1 \times \cdots \times D_k$, each element of which is values that the variables can not take simultaneously. This set is referred to as the no-good set (i.e., an assignment set that is not contained in any solution.)

A solution to a CSP requires the assignment of values to each of the variables from their domains such that all the constraints on the variables are satisfied. In this paper, attention is focused on binary CSPs, where all constraints are binary; that is, they are based on the Cartesian product of the domains of two variables. However, any nonbinary CSP can theoretically be converted to a binary CSP [8, 9]. The structure of a binary CSP can be better visualized by a graph $G(V, E)$ where the set of vertices V corresponds to the variables and each edge $(X_i, X_j) \in E$ represents a constraint connecting the pair of variables involved in this constraint. The CSP in its general form is NP-complete [10] and has been extensively studied due to its simplicity and applicability [7]. The simplicity of the problem coupled with its intractability makes it an ideal platform for exploring new algorithmic techniques. This has led to the development of several algorithms for solving CSPs which usually fall into two main categories: systematic algorithms and local search algorithms.

3. A Brief Survey of Methods

Systematic search algorithms rely on a systematic way in their exploration of the search space. These methods [11–16] aim at exploring the entire solution space using tree search algorithms. The two main components of a tree search are the way to go forward, that is, which decision is taken at which point of the search and the way to go backwards, that is, the backtracking strategy that defines how the algorithm will behave when an inconsistency is detected. In practice, methods based on systematic tree search may fail to solve large and complex CSPs instances because the computing time required may become prohibitive. For instance, a CSP with n variables, each with a domain of size m, makes the search space which is to be explored proportional to $O(m^n)$, that is, exponential in the number of variables. Most searches that come up in CSPs occur over spaces that are far too large to be searched exhaustively. One way to overcome the combinatorial explosion is to give up completeness. Stochastic local search (SLS) algorithms are techniques which use this strategy and gained popularity due to their conceptual simplicity and good performance. These methods start with an initial assignment of values to variables randomly or heuristically generated. During each iteration, a new solution is selected from the neighborhood of the current one by performing a move. A move might consist in changing the value of one randomly selected variable. Choosing a good neighborhood and a method for searching it is usually guided by intuition, because very little theory is available as a guide. If the new solution provides a better value in light of the objective function, the new solution becomes the current one. In order to avoid premature convergence, SLS methods resort to some sort of randomization (noise probability) to avoid local minima and to better explore the search space. The search is iterated until a termination criterion is reached. Most algorithms applied to CSPs use the so-called 1-exchange neighborhood under which two solutions are direct neighbors if, and only if, they differ at most in the value assigned to one variable. A basis for many SLS algorithms is the minimum conflict heuristic MCH [17]. MCH iteratively modifies the assignment of a single variable in order to minimize the number of violated constraints. Since the introduction of MCH there have been a large number of local search heuristics proposed to tackle CSPs. Several representative state-of-the-art SLS in the literature include the break method for escaping from local minima [18], various enhanced MCH (e.g., randomized iterative improvement of MCH called WMCH [19], MCH with tabu search [20, 21]), and a large body of work on evolutionary algorithms for CSPs [22–26] for interested readers. Weights-based algorithms have been advocated by the intuition that, by introducing weights on variables or constraints, local minima can be avoided and the search process can learn to distinguish between critical and less critical constraints. Methods belonging to this category include genet [27], guided local search [28], discrete Lagrangian search [29], the exponentiated subgradient [30], the scaling and probabilistic smoothing [31], evolutionary algorithms combined with stepwise adaptation of weights [32–34], methods based on dynamically adapting weights on variables [35], or both (i.e., variables and constraints) [36]. Weighting schemes have been also combined with systematic methods to reduce the size of tree search methods and consequently speeding up the solving time [37–39]. Recently, an improved version of the Squeaky Wheel Optimization (SWO) [40] originated in [41] has been proposed for the scheduling problem. In SWO, a greedy algorithm is used to construct an initial solution which is then analyzed in order to identify those tasks that if improved are likely to improve the objective

function score. The improved version provides additional postprocessing transformations to explore the neighborhood enhanced with a stochastic local search algorithm. Methods based on large neighborhood search have recently attracted several researchers for solving the CSP [42]. The central idea is to reduce the size of local search space relying on a continual relaxation (removing elements from the solution) and reoptimization (reinserting the removed elements). Systematic methods exhibit poor performance on large problems because bad decisions made early in the search persist for exponentially long times. In contrast, stochastic local search methods replace systematicity with stochastic techniques for diversifying the search. However, the lack of systematicity makes remembering the history of past states problematic. To this end, hybrid search methods offering desirable aspects of both systematic methods and local search methods are becoming more and more popular and interested readers may refer to [43–45] to get a deeper understanding on these mixed methods.

4. Variable Depth Search Algorithm

Traditional local search algorithms for solving CSP problems start from an initial solution s and repeat replacing s with a better solution in its neighborhood $N(s)$ until no better solution is found in $N(s)$, where $N(s)$ is a set of solutions obtained from s by updating the value of one selected variable. A solution $s*$ is called locally optimal if no better solution exists in $N(s*)$. The algorithm proposed in this paper belongs to the class of variable depth search algorithms where an existing solution is not modified just by making a change to a single variable; instead, the changes affect as many variables as possible when moving from one solution to another. The algorithm is inspired from the famous Kerninghan-Lin algorithm used for solving the graph partitioning problem [46] and the traveling salesman problem [47]. The idea is to replace the search for one favorable move (i.e., the update of one variable) by a search for a favorable sequence of moves (i.e., the update of a series of variables) using the criterion of score to guide the search. The different steps of the algorithm are described in Algorithm 1.

(i) Random-initial-solution (): the algorithm starts building an initial solution. The initial solution is simply constructed by assigning to each variable X_i a random value v_i from D_{X_i} (Line 5 of Algorithm 1). Based on these values, the status of each constraint is set to either violated or nonviolated.

(ii) Assign-Initial-Weights (): during this step the algorithm assigns a fixed amount of weight equal to 1 across all the constraints (Line 6 of Algorithm 1). The distribution of weights to constraints is a key factor to the success of the algorithm. During the course of the search, the algorithm forces hard constraints (i.e., those with large weights) to be satisfied thereby preventing the algorithm at a later stage from getting stuck at a local optimum.

(iii) Stopping criterion: the outer loop (Line 7 of Algorithm 1) determines the stopping criterion met by the algorithm. The algorithm stops if a solution has been found (i.e., all the constraints are satisfied) or if a time limit has been reached.

(iv) Random-selected-variable (): a starting random variable from which the searching process begins is selected and added to the set T (Lines: 9, 10, and 11 of Algorithm 1).

(v) Inner loop: the inner loop (Lines: 12, 13, 14, 15, 16, 17, and 18 of Algorithm 1) proceeds by repeatedly selecting for each variable X_i removed from the set T, the value v_{best} from its domain D_{X_i} producing the highest score. Given the choice between several equally high scores, the algorithm picks one value at random. The score of a variable $X_i^{v_j}$ is defined as the increase (or decrease in the number of violated constraints) in the number of nonviolated constraints if X_i is assigned the value v_j. The score is given by

$$\text{Score}\left(X_i^{v_{\text{best}}}\right) = \text{New}\left(X_i^{v_{\text{best}}}\right) - \text{Current}\left(X_i^{v_{\text{current}}}\right), \quad (1)$$

$$\begin{aligned}&\text{New}\left(X_i^{v_{\text{best}}}\right)\\ &= \sum_{X_j \in \text{Neigh}(X_i)}^{|\text{Neigh}(X_i)|} \Omega\left(X_i, X_j\right) * \Phi\left(X_i^{v_{\text{best}}}, X_j^{v_{\text{current}}}\right),\\ &\text{Current}\left(X_i^{v_{\text{current}}}\right)\\ &= \sum_{X_j \in \text{Neigh}(X_i)}^{|\text{Neigh}(X_i)|} \Omega\left(X_i, X_j\right) * \Phi\left(X_i^{v_{\text{current}}}, X_j^{v_{\text{current}}}\right).\end{aligned} \quad (2)$$

Equations (2) calculates the sum of the weights of the satisfied constraints the variable X_i is involved with. $\Omega(X_i, X_j)$ denotes the weight of the constraint connecting X_i and X_j while the second term returns the value of 1 if the constraint is satisfied and 0 otherwise. Thus, after the selection of v_{best} and inserting $X_i^{v_{\text{best}}}$ into the set M_{Best}, the status (i.e., violated or nonviolated) of the constraints for the neighboring variables of X_i is updated. Consider the following:

(i) Highest cumulative score: an iteration of the algorithm terminates when the set T becomes empty. In this way, a sequence of scores with corresponding variables and their selected values is formed. Thereafter, the algorithm identifies the subset of variables having the highest cumulative score (HCS) (Line 19 of Algorithm 1). The identification of this subset is equivalent to choosing k so that HCS(k) in (3) is maximum, where $S_{X_i}^{v_{\text{best}}}$ represents the score of the variable X_i corresponding to the value v_{best}. Finding k is the same as solving the maximum subarray problem introduced for the first time in [48]. The problem is usually solved using Kadane's algorithm [49] which simply accumulates a partial sum and

input: Problem Instance
output: Number of satisfied constraints
(1) **begin**
(2) Let $Neigh(X_i) = \{X_j \mid (X_i, X_j) \in E, i = 1, \ldots, n, j = 1, \ldots, n\}$;
(3) Let $X_i^{v_j}$ denotes the assignment of the value v_i from D_{X_i} to X_i;
(4) Let $M_{\text{Best}} = \{X_i^{v_{\text{best}}} \mid Score(X_i^{v_{\text{best}}}) >= Score(X_i^{v_j}), i = 1, \ldots, n, j = 1, \ldots, |D_{X_i}|\}$
(5) Random-Initial-Solution ();
(6) Assign-Initial-Weights ();
(7) **while** (!*stop*) **do**
(8) $M_{\text{Best}} = \emptyset$;
(9) $T = \emptyset$;
(10) $X_i \leftarrow$ Random-Selected-Variable ();
(11) $T \leftarrow T \cup \{X_i\}$;
(12) **while** ($T \neq \emptyset$) **do**
(13) $T \setminus \{X_k\} \leftarrow$ Remove a random variable X_k from T;
(14) $X_k^{v_{\text{best}}} \leftarrow$ Assign the value v_{best} to X_k producing the highest score;
(15) $T \leftarrow T \cup \{X_j | X_j \in Neigh(X_i) \wedge X_j \notin T\}$;
(16) $M_{\text{Best}} \leftarrow M_{\text{Best}} \cup \{X_k^{v_{\text{best}}}\}$;
(17) Update-Score of Neigh (X_k);
(18) **end**
(19) Identify the set of variables with the highest cumulative score (HCS):
(20) $\text{HCS}(k) = \sum_{i=1, X_i^{v_{\text{best}}} \in M_{\text{best}}}^{k} (S_{X_i^{v_{\text{best}}}})$;
(21) **if** ($\text{HCS}(k) \geq 0$) **then**
(22) Assign all the variables up to the index k with their new best values;
(23) **else**
(24) Assign the variable at the index 1 with its new best value;
(25) **end**
(26) Adjust-Weights ();
(27) **end**
(28) **end**

Algorithm 1: VNS-CSP.

updates the optimal range when this partial sum becomes larger than the global sum. If HCS ≥ 0, the solution is updated by substituting all the variables up to the index k with their new values; otherwise the update is restricted to just the first variable (index 1) (Lines: 20, 21, 22, 23, and 24 of Algorithm 1):

$$\text{HCS}(k) = \sum_{i=1, X_i^{v_{\text{best}}} \in M_{\text{best}}}^{k} S_{X_i^{v_{\text{best}}}}. \quad (3)$$

(ii) Adjust-Weights: finally, the algorithm proceeds with the weighting process divided into two distinct steps (Line 25 of Algorithm 1). The weights of each newly violated constraint are then increased by one, whereas the newly satisfied constraints will have their weights decreased by one before another round of the algorithm is repeated or the stopping criterion is reached. This weighting procedure is the same as the one adopted in [18].

5. Experimental Results

5.1. Test Instances. The performance of the metaheuristic (VNS-CSP) has been tested on hard random CSP problems taken from Lecoutres benchmark [50] under the name RB-Model. This model enjoys several features that makes it of a great theoretical and practical interest [51]. Tables 1 and 2 show the list of problem instances used in the experiments. The list contains 8 classes of problems each of which is composed of 5 instances, giving a total of 40 instances. Table 1 shows the list of solved hard problems, while Table 2 refers to those problems that remain challenging for most solvers. They are all located in the exact phase transition point [52] and the hardness of solving these instances grows exponentially with the number of variables. The first column denotes the number of variables, the second column the domain size of the each variable, and the third column the number of constraints; the fourth column specifies the combination of values not allowed (no-good) and the last column shows whether the instance has already been solved by an existing solver. All the benchmark instances used in this experiment are satisfiable instances. Each problem instance was run 100 times (i.e., each run is performed with a different seed) with a cut-off parameter (max-time) set to 15 minutes. The tests were carried out on a DELL machine with 800 MHz CPU and 2 GB of memory. The code was written in C and compiled with the GNU C compiler version 4.6.

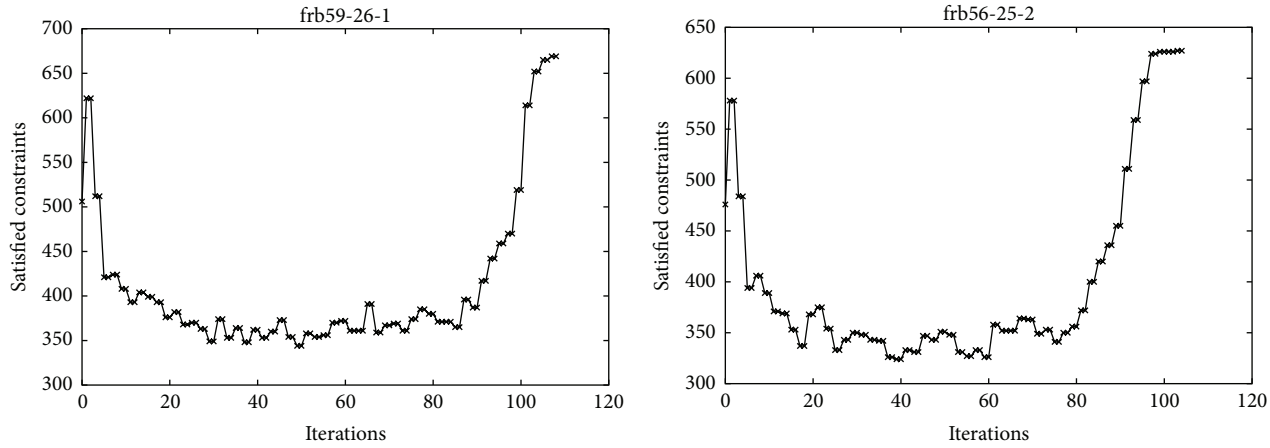
Table 1: Solvable instances.

Instance	Variables	Values	Constraints	No-good	Solved
frb30-15-1.csp	30	15	284	56	Yes
frb30-15-2.csp	30	15	284	56	Yes
frb30-15-3.csp	30	15	284	56	Yes
frb30-15-4.csp	30	15	284	56	Yes
frb30-15-5.csp	30	15	284	56	Yes
frb35-15-1.csp	35	17	346	72	Yes
frb35-15-2.csp	35	17	346	72	Yes
frb35-15-3.csp	35	17	346	72	Yes
frb35-15-4.csp	35	15	346	72	Yes
frb35-15-5.csp	35	15	346	72	Yes
frb40-19-1.csp	40	19	410	90	Yes
frb40-19-2.csp	40	19	410	90	Yes
frb40-19-3.csp	40	19	410	90	Yes
frb40-19-4.csp	40	19	410	90	Yes
frb40-19-5.csp	40	19	410	90	Yes
frb45-21-1.csp	45	21	476	110	Yes
frb45-21-2.csp	45	21	476	110	Yes
frb45-21-3.csp	45	21	476	110	Yes
frb45-21-4.csp	45	21	476	110	Yes
frb45-21-5.csp	45	21	476	110	Yes
frb53-24-3.csp	53	24	585	144	Yes

Table 2: Benchmark instances: unsolvable instances.

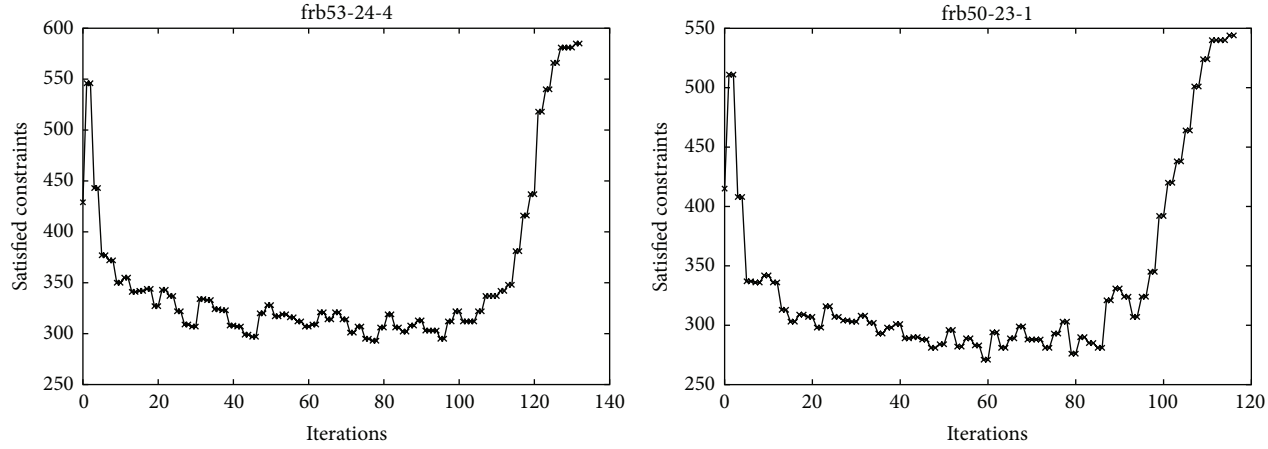
Instance	Variables	Values	Constraints	No-good	Solved
frb50-23-1.csp	50	23	544	132	**No**
frb50-23-2.csp	50	23	544	132	**No**
frb50-23-3.csp	50	23	544	132	**No**
frb50-23-4.csp	50	23	544	132	**No**
frb50-23-5.csp	50	23	544	132	**No**
frb53-24-1.csp	53	24	585	144	**No**
frb53-24-2.csp	53	24	585	144	**No**
frb53-24-4.csp	53	24	585	144	**No**
frb53-24-5.csp	53	24	585	144	**No**
frb56-25-1.csp	56	25	627	156	**No**
frb56-25-2.csp	56	25	627	156	**No**
frb56-25-3.csp	56	25	627	156	**No**
frb56-25-4.csp	56	25	627	156	**No**
frb56-25-5.csp	56	25	627	156	**No**
frb59-26-1.csp	59	26	669	169	**No**
frb59-26-2.csp	59	26	669	169	**No**
frb59-26-3.csp	59	26	669	169	**No**
frb59-26-4.csp	59	26	669	169	**No**
frb59-26-5.csp	59	26	669	169	**No**

5.2. Algorithm's Behavior. The plots depicted in Figures 1 and 2 show the evolution of the mean satisfied number of constraints as a function of the number of iterations for 4 hard problems that remain difficult for most solvers. These plots have been selected as they represent the general trend observed on all the problem instances. Investigating the trends of the algorithm from the plots suggests the presence of three different distinct phases. The first phase corresponds to the first iteration of the algorithm where all the constraints are assigned a weight equal to 1. This similar weight provides all the constraints with equal chances for being satisfied. In all the studied cases, the curves have a tendency to go uphill showing an improvement in the number of satisfied constraints. The second phase which takes most of the time corresponds to a diversification stage. During this second phase, the weights assigned to various constraints alter after each iteration depending on the status of the constraints (i.e., satisfied or unsatisfied) forcing the algorithm to favor the satisfaction of hard constraints (i.e, constraints with higher weights). This weighting of constraints results in worsening the quality of the solution by falling drastically during early stages of this phase (on average between 33% and 53%) and continues to exhibit a varying increasing decline rate over time before the curves start moving uphill marking the start of the intensification phase. This phase which tends to be of short duration compared to the diversification phase is characterized by the absence of downhill moves. A downhill move occurs when the set of changes determined by the algorithm reduces the number of satisfied constraints. During the intensification phase, the algorithm intensifies the search around promising areas of the search space making the number of satisfied constraints to climb sharply until all the constraints of the problem are satisfied. The termination of the diversification phase ensures that each constraint relating at most two variables is assigned an ideal weight expressing its relative hardness taking into account the values assigned to its relating variables and the values of the variables defining the neighboring constraints. This ideal weight leads the system to enter a state of balance that is required for the intensification phase to be triggered leading the algorithm to easily reach the solution of the problem. Figures 3 and 4 show the evolution of the number of satisfied constraints and the sum of weights of satisfied constraints through the diversification and intensification phases, respectively. Figure 3 reveals that improving the sum of weights of satisfied constraints does not necessarily imply an increase in the number of satisfied constraints. Satisfying constraints with large weights may introduce a new set of unsatisfied constraints leading to a further decrease in the number of satisfied constraints. Another interesting remark to be drawn from this plot is the ability of the algorithm to escape from the so-called plateau regions or local optima. Plateaus represent regions of the search space containing states with only equal or disimproving costs leaving the best solution unchanged. Figure 4 shows a continuous improvement of the two curves during the intensification phase until the solution of the problem is reached. Figure 5 shows the impact of weighting and nonweighting strategies on the algorithm's convergence. The plot illustrates the easiness encountered by the algorithm without the weighting mechanism in improving the number of satisfied constraints during the first iterations of the algorithm (up to 96% of the constraints are satisfied) before getting permanently stuck in long plateau regions or a local maximum leading to a premature convergence due to its greedy bias. The superior performance of the algorithm is



(a) (b)

Figure 1: Evolution of the number of satisfied constraints: (a) frb59-26-1 and (b) frb56-25-2.



(a) (b)

Figure 2: Evolution of the number of satisfied constraints: (a) frb53-24-4 and (b) frb50-23-1.

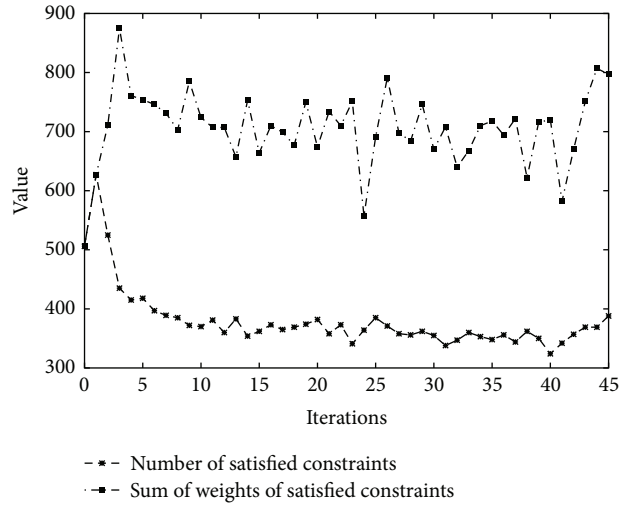


Figure 3: Evolution of the number of satisfied constraints and the sum of weights of satisfied constraints for frb59-26-3 during the diversification phase.

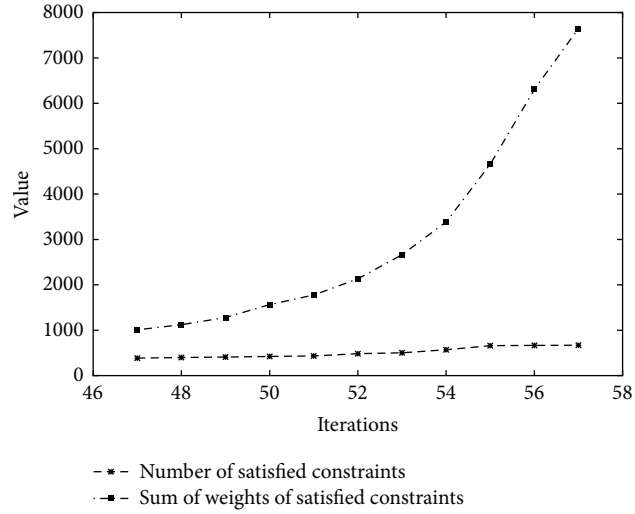


Figure 4: Evolution of the number of satisfied constraints and the sum of weights of satisfied constraints for frb59-26-3 during the intensification phase.

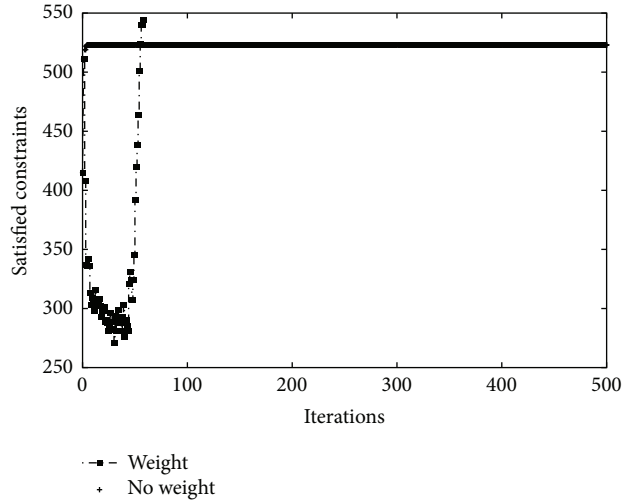


Figure 5: Impact of the nonweight mechanism on the algorithm's convergence for frb50-23-3.

further made evident by looking at Table 3 which presents the results for already solved problem instances and unsolvable problem instances (instances in bold) that still present a challenge for all existing solvers. The results illustrate the performance of the algorithm reflecting its success ratio (i.e., defined as the ratio of successful runs with respect to the total number of runs) and the amount of time taken to reach the solution. From these results, the algorithm has a very good reliability (the success ratio is 100%). In terms of speed, VNS-CSP reaches the solutions in short computational times. Hence much of the difference in the run time (max-min) is due to the different random initial solutions and to the first chosen random variable that initiates the searching process.

5.3. Comparison with State-of-the-Art Solvers. Tables 4–7 compare the time (i.e., the average time over 100 runs) required for different state-of-the-art solvers relative to that required by VNS-CSP. All these solvers are complete (i.e., systematic) solvers. The dash symbol means that the solver could not find the solution after 30 minutes. The first row on each table refers to the metaheuristic proposed in this work VNS-CSP. In all cases, the proposed metaheuristic remains the fastest of them all. The time of the proposed metaheuristic ranges from 10% to 90% of the time of the best solver and from 5 to several hundred times faster than the slowest solver. Table 8 compares VNS-CSP against two variants of Ant Colony Optimization (ACO) algorithms and tabu search. This table is extracted from [53]. The first column shows the 8-class problems each of which is composed of 5 different instances. The second and third columns show the results of the two variants of ACO. The first number represents the number of solved instances, while the number in bracket gives the average CPU time on 3 GHz Intel Xeon. The last column shows the result of VNS-CSP. The time in bracket is the average time taken on DELL machine with 800 MHz CPU. This table is only meant as a rough guide since VNS-CSP and the other algorithms are run on different machines. The table shows that the two variants of ACO and tabu are outperformed by VNS-CSP. VNS-CSP solved all the instances, while ACO-vertex has been able to solve 29 out of 40, ACO-clique 28 out of 40, and tabu 36 out of 40. Comparing the time of the different algorithms, VNS-CSP is the one requiring the least amount of time.

Table 3: Benchmark instances: all unsolved instances solved.

	Execution time and success ratio			
Instances	Min (sec)	Max (sec)	Mean (sec)	Success ratio
frb30-15-1.csp	0.18	0.20	0.19	100%
frb30-15-2.csp	0.15	0.17	0.16	100%
frb30-15-3.csp	0.16	0.17	0.17	100%
frb30-15-4.csp	0.18	0.19	0.19	100%
frb30-15-5.csp	0.14	0.23	0.20	100%
frb35-17-1.csp	0.22	0.27	0.25	100%
frb35-17-2.csp	0.29	0.42	0.31	100%
frb35-17-3.csp	0.21	0.25	0.24	100%
frb35-17-4.csp	0.20	0.23	0.22	100%
frb35-17-5.csp	0.27	0.30	0.28	100%
frb40-19-1.csp	0.37	0.42	0.41	100%
frb40-19-2.csp	0.34	0.38	0.36	100%
frb40-19-3.csp	0.41	0.43	0.42	100%
frb40-19-4.csp	0.36	0.42	0.38	100%
frb40-19-5.csp	0.39	0.41	0.40	100%
frb45-21-1.csp	0.50	0.54	0.52	100%
frb45-21-2.csp	0.52	0.55	0.54	100%
frb45-21-3.csp	0.60	0.66	0.62	100%
frb45-21-4.csp	0.54	0.56	0.54	100%
frb45-21-5.csp	0.53	0.58	0.56	100%
frb50-23-4.csp	0.78	0.86	0.81	100%
frb53-24-2.csp	0.87	0.92	0.89	100%
frb50-23-1.csp	0.72	0.84	0.79	100%
frb50-23-2.csp	0.73	0.76	0.75	100%
frb50-23-3.csp	0.86	0.93	0.88	100%
frb50-23-5.csp	0.77	0.81	0.80	100%
frb53-24-1.csp	0.95	1.01	0.98	100%
frb53-24-3.csp	0.88	0.95	0.87	100%
frb53-24-4.csp	1.05	1.09	1.07	100%
frb53-24-5.csp	1.15	1.18	1.17	100%
frb56-25-1.csp	1.20	1.22	1.21	100%
frb56-25-2.csp	0.99	1.03	1.01	100%
frb56-25-3.csp	1.01	1.05	1.02	100%
frb56-25-4.csp	0.98	0.99	0.99	100%
frb56-25-5.csp	1.09	1.14	1.11	100%
frb59-26-1.csp	1.24	1.33	1.30	100%
frb59-26-2.csp	1.22	1.26	1.24	100%
frb59-26-3.csp	1.31	1.34	1.32	100%
frb59-26-4.csp	1.30	1.32	1.31	100%
frb59-26-5.csp	1.23	1.27	1.24	100%

Table 4: Comparing various solvers: frb30 instances.

Solver	frb30-15-1	frb30-15-3	frb30-15-4	frb30-15-5
VNS-CSP	**0.19**	**0.17**	**0.19**	**0.20**
Abscon112v4	0.91	1.41	1.33	1.65
Abscon 112v4ESAC	0.88	1.36	1.35	1.67
Bpsolver09	0.67	2.49	2.59	1.3
Choco2.1.1	1.48	2.21	2.55	1.58
Choco2.1.1b	1.82	1.14	2.33	0.90
Concrete	1.89	1.25	2.20	1.75
Concrete DC	2.59	2.94	3.30	2.41
Conquer	2.45	2.77	1.00	1.61
Mistral	0.21	0.22	0.26	0.08
pcs	3.05	2.51	1.93	1.07
pcs-restart	4.92	0.55	2.55	0.70
SAT4JCSP	3.74	4.81	5.50	3.95
Sugarv1.14.6 + minisat	3.3	1.17	1.97	1.15
Sugarv1.14.6 + picosat	2.11	1.47	1.81	1.57

Table 5: Comparing various solvers: frb35 instances.

Solver	frb35-17-1	frb35-17-2	frb35-17-4
VNS-CSP	**0.25**	**0.31**	**0.22**
Abscon112v4	4.37	6.71	3.54
Abscon 112v4ESAC	3.92	6.38	3.87
Bpsolver09	16.81	67.03	25.30
Choco2.1.1	4.02	35.90	8.47
Choco2.1.1b	5.40	16.27	1.70
Concrete	5.20	10.86	4.49
Concrete DC	6.68	7.05	5.83
Conquer	2.74	14.73	5.57
Mistral	0.64	3.08	0.25
pcs	42.05	27.39	8.15
pcs-restart	47.99	27.07	1.94
SAT4JCSP	51.20	212.57	43.08
Sugarv1.14.6 + minisat	13.69	25.35	4.17
Sugarv1.14.6 + picosat	14.89	4.72	14.37

6. Conclusions

This paper proposes a variable depth search algorithm for the CSP problem. The heart of the metaheuristic relies on a combination between an adaptive weighting strategy on the constraint weights and a greedy search. This combination proved to be an excellent mechanism to guide the search in order to achieve a suitable trade-off between intensification and diversification. The proposed metaheuristic has been experimentally evaluated on hard random CSP problems belonging to RB-Model. The difficulty of solving some of these problems by state-of-the-art solvers highlights the capabilities of the proposed metaheuristic. Indeed, the experimental results have been very positive, solving all unsolvable instances in very short computational times. Most

Table 6: Comparing various solvers: frb40 instances.

Solver	frb40-19-1	frb40-19-4	frb40-19-5
VNS-CSP	**0.41**	**0.38**	**0.40**
Abscon112v4	1.30	82.98	37.29
Abscon112v4ESAC	1.32	79.60	36.6
Bpsolver09	361.45	137.45	199.04
Choco2.1.1	25.43	24.33	161.97
Choco2.1.1b	25.43	24.33	161.97
Concrete	15.36	87.77	177.15
Concrete DC	21.27	114.80	89.01
Conquer	21.66	9.49	94.78
Mistral	2.15	7.73	63.46
pcs	1758.00	—	—
pcs-restart	—	—	1107.76
SAT4JCSP	272.64	500.71	10.49
Sugarv1.14.6 + minisat	48.30	186.73	—
Sugarv1.14.6 + picosat	41.52	364.37	297.49

Table 7: Comparing various solvers: frb45 and frb53 instances.

Solver	frb45-21-2	frb45-21-4	frb45-21-5	frb53-24-3
VNS-CSP	**0.54**	**0.54**	**0.56**	**0.87**
Abscon112v4	275.31	478.01	1228.41	1342
Abscon112v4ESAC	284.00	441.30	1194.47	245.31
Bpsolver09	—	—	—	—
Choco2.1.1	1305.51	65.05	1635.94	—
Choco2.1.1b	423.62	217.20	89.18	—
Concrete	394.55	330.23	672.39	—
Concrete DC	498.34	359.80	1023.49	—
Conquer	800.71	716.26	878.97	—
Mistral	224.39	66.05	121.84	—
pcs	—	—	—	—
pcs-restart	—	—	—	—
SAT4JCSP	—	—	—	—
Sugarv1.14.6 + minisat	56.95	—	1114.95	—
Sugarv1.14.6 + picosat	795.52	—	—	—

Table 8: Comparing VNS-CSP with tabu and ACO metaheuristics.

Instances	ACO-SSP (vertex)	ACO-SSP (clique)	Tabu	VNS-CSP
frb30-15	5 (**0.4**)	5 (**1.3**)	5 (**0.5**)	5 (**0.18**)
frb35-17	5 (**3.0**)	5 (**5.0**)	5 (**0.9**)	5 (**0.26**)
frb40-19	5 (**7.0**)	5 (**103.5**)	5 (**9.1**)	5 (**0.39**)
frb45-21	5 (**467.7**)	5 (**354.1**)	5 (**43.4**)	5 (**0.56**)
frb50-23	3 (**430.4**)	3 (**680.5**)	4 (**9.9**)	5 (**0.80**)
frb53-24	3 (**105.7**)	3 (**530.8**)	4 (**291.6**)	5 (**0.99**)
frb56-25	2 (**535.8**)	2 (**170.2**)	4 (**329.3**)	5 (**1.01**)
frb59-26	1 (**63.6**)	0 (—)	4 (**523.7**)	5 (**1.28**)

metaheurirics have a predefined set of parameters that has to be calibrated with respect to the problem at hand. This parameter tuning which becomes a tedious task as the number of parameter increases plays a significant impact on the solving progress and therefore the solution quality. What distinguishes the proposed metaheuristic from state-of-the-art techniques is the absence of parameter tuning making it highly suitable in practical scenarios.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] M. Dincbas and H. Simonis, "APACHE—a constraint based, automated stand allocation system," in *Proceedings of the Advanced Software Technology in Air Transport (ASTAIR '91)*, pp. 267–282, Royal Aeronautical Society, London, UK, 1991.

[2] J. Bellone, A. Chamard, and C. Pradelles, "PLANE -an evolutive planning system for aircraft production," in *Proceedings of the 1st International Conference on Practical Application of Prolog*, 1992.

[3] H. Simonis and P. Charlier, "COBRA—a system for train crew Scheduling," in *Proceedings of the Workshop on Constraint Programming and Large Scale Combinatorial Optimization (DIMACS '98)*, Rutgers University, New Brunswick, NJ, USA, September 1998.

[4] G. Baues, P. Kay, and P. Charlier, " Constraint based resource allocation for airline crew management," in *PRoceedings of the ATTIS*, Paris, France, April 1994.

[5] R. Amadini, I. Sefrioui, J. Mauro, and M. Gabbrielli, "A constraint-based model for fast post-disaster emergency vehicle routing," *International Jorunal of Interactive Multimedia and Artificial Intelligence*, vol. 2, no. 4, pp. 67–75, 2013.

[6] K. Kinoshita, K. Iizuka, and Y. Iizuka, "Effective disaster evacuation by solving the distributed constraint optimization problem," in *Proceedings of the 2nd IIAI International Conference on Advanced Applied Informatics (IIAIAAI '13)*, pp. 399–400, Los Alamitos, Calif, USA, September 2013.

[7] F. Rossi, P. V. Beek, and T. Walsh, *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*, Elsevier Science, New York, NY, USA, 2006.

[8] R. Dechter and J. Pearl, "Tree clustering for constraint networks," *Artificial Intelligence*, vol. 38, no. 3, pp. 353–366, 1989.

[9] F. Rossi, C. Petri, and V. Dhar, "On the equivalence of constraint satisfaction problems," in *Proceedings of the European Conference on Artificial Intelligence (ECAI '90)*, pp. 550–556, 1990.

[10] A. K. Mackworth, "Consistency in networks of relations," *Artificial Intelligence*, vol. 8, no. 1, pp. 99–118, 1977.

[11] R. Dechter and D. Frost, "Backjump-based backtracking for constraint satisfaction problems," *Artificial Intelligence*, vol. 136, no. 2, pp. 147–188, 2002.

[12] N. Jussien, G. Rochart, and X. Lorca, "Choco: an open source java constraint programming Library," in *Proceedings of the Workshop on Open-Source Software for Integer and Contraint Programming (OSSICP '08)*, pp. 1–10, Paris, France, June 2008.

[13] S. Merchez, C. Lecoutre, and F. Boussemart, "AbsCon: a prototype to solve CSPs with abstraction," in *Principles and Practice of Constraint Programming—CP 2001*, vol. 2239 of *Lecture Notes in Computer Science*, pp. 730–744, Springer, Berlin, Germany, 2001.

[14] J.-C. Régin, "AC-*: a configurable, generic and adaptive arc consistency algorithm," in *Principles and Practice of Constraint Programming—CP 2005*, vol. 3709 of *Lecture Notes in Computer Science*, pp. 505–519, Springer, Berlin, Germany, 2005.

[15] D. Sabin and E. C. Freuder, "Contradicting conventional wisdom in constraint satisfaction," in *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI '94)*, pp. 125–129, Amsterdam, The Netherlands, August 1994.

[16] N. Tamura, A. Taga, S. Kitagawa, and M. Banbara, "Compiling finite linear CSP into SAT," *Constraints*, vol. 14, no. 2, pp. 254–272, 2009.

[17] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems," *Artificial Intelligence*, vol. 58, no. 1–3, pp. 161–205, 1992.

[18] P. Morris, "The breakout method for escaping from local minima," in *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI '93)*, pp. 40–45, 1993.

[19] R. J. Wallace and E. C. Freuder, "Heuristic methods for over-constrained constraint satisfaction problems," in *Over-Constrained Systems*, vol. 1106 of *Lecture Notes in Computer Science*, pp. 207–216, Springer, Berlin, Germany, 1996.

[20] P. Galinier and J.-K. Hao, "Tabu search for maximal constraint satisfaction problems," in *Principles and Practice of Constraint Programming-CP97*, vol. 1330 of *Lecture Notes in Computer Science*, pp. 196–208, Springer, Berlin, Germany, 1997.

[21] T. Stützle, *Local search algorithms for combinatorial problems—analysis, improvements, and new applications [Ph.D. thesis]*, TU Darmstadt, FB Informatik, Darmstadt, Germany, 1998.

[22] N. Bacanin and M. Tuba, "Artificial bee colony (ABC) algorithm for constrained optimization improved with genetic operators," *Studies in Informatics and Control*, vol. 21, no. 2, pp. 137–146, 2012.

[23] M. R. Bonyadi, X. Li, and Z. Michalewicz, "A hybrid particle swarm with velocity mutation for constraint optimization problems," in *Proceedings of the 15th Genetic and Evolutionary Computation Conference (GECCO '13)*, pp. 1–8, ACM, New York, NY, USA, July 2013.

[24] D. Curran, E. Freuder, and T. Jansen, "Incremental evolution of local search heuristics," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference (GECCO '10)*, pp. 981–982, ACM, New York, NY, USA, July 2010.

[25] S. Voß, "Meta-heuristics: state of the art," in *Local Search for Planning and Scheduling*, vol. 2148 of *Lecture Notes in Computer Science*, pp. 1–23, Springer, Berlin, Germany, 2001.

[26] Y. Zhou, G. Zhou, and J. Zhang, "A hybrid glowworm swarm optimization algorithm for constrained engineering design problems," *Applied Mathematics and Information Sciences*, vol. 7, no. 1, pp. 379–388, 2013.

[27] A. Davenport, E. Tsang, C. J. Wang, and K. Zhu, "Genet: a connectionist architecture for solving constraint satisfaction problems by iterative improvement," in *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI '94)*, pp. 325–330, Seattle, Wash, USA, August 1994.

[28] C. Voudouris and E. P. K. Tsang, "Guided local search," in *Handbook of Metaheuristics*, vol. 57 of *International Series in Operation Research and Management Science*, pp. 185–218, Kluwer Academic Publishers, Boston, Mass, USA, 2003.

[29] Y. Shang and B. W. Wah, "A discrete Lagrangian-based global-search method for solving satisfiability problems," *Journal of Global Optimization*, vol. 12, no. 1, pp. 61–99, 1998.

[30] D. Schuurmans, F. Southey, and R. C. Holte, "The exponentiated subgradient algorithm for heuristic Boolean programming," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI '01)*, pp. 334–341, Morgan Kaufmann, San Francisco, Calif, USA, August 2001.

[31] F. Hutter, D. A. D. Tompkins, and H. H. Hoos, "Scaling and probabilistic smoothing: efficient dynamic local search for SAT," in *Principles and Practice of Constraint Programming—CP 2002*, vol. 2470 of *Lecture Notes in Computer Science*, pp. 233–248, Springer, Berlin, Germany, 2002.

[32] D. A. H. Amante and H. T. Marin, "Adaptive penalty weights when solving congress timetabling," in *Advances in Artificial Intelligence—IBERAMIA 2004*, vol. 3315 of *Lecture Notes in Computer Science*, pp. 144–153, Springer, Berlin, Germany, 2004.

[33] M. R. Karim, "A new approach to constraint weight learning for variable ordering in CSPs," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 2716–2723, Beijing, China, July 2014.

[34] R. Shalom, M. Avigal, and R. Unger, "A conflict based SAW method for constraint satisfaction problems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 373–380, IEEE, Trondheim, Norway, May 2009.

[35] W. Pullan, F. Mascia, and M. Brunato, "Cooperating local search for the maximum clique problem," *Journal of Heuristics*, vol. 17, no. 2, pp. 181–199, 2011.

[36] S. Fang, Y. Chu, K. Qiao, X. Feng, and K. Xu, "Combining edge weight and vertex weight for minimum vertex cover problem," in *Frontiers in Algorithmics: 8th International Workshop, FAW 2014, Zhangjiajie, China, June 28–30, 2014. Proceedings*, vol. 8497 of *Lecture Notes in Computer Science*, pp. 71–81, Springer, 2014.

[37] F. Boussemart, F. Hemery, C. Lecoutre, and L. Sais, "Boosting systematic search by weighting constraints," in *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI '04)*, pp. 146–150, August 2004.

[38] M.-J. Huguet, P. Lopez, and W. Karoui, "Weight-based heuristics for constraint satisfaction and combinatorial optimization problems," *Journal of Mathematical Modelling and Algorithms*, vol. 11, no. 2, pp. 193–215, 2012.

[39] M. Mouhoub and B. Jafari, "Heuristic techniques for variable and value ordering in CSPs," in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11)*, pp. 457–464, ACM, Dublin, Ireland, July 2011.

[40] A. Alexiadis and J. Refanidis, "Post-optimizing individual activity plans through local search," in *Proceedings of the 8th Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems (COPLAS '13)*, pp. 7–15, 2013.

[41] D. Joslin and D. P. Clements, "Squeaky wheel optimization," *Journal of Artificial Intelligence Research*, vol. 10, pp. 353–373, 1999.

[42] H.-J. Lee, S.-J. Cha, Y.-H. Yu, and G.-S. Jo, "Large neighborhood search using constraint satisfaction techniques in vehicle routing problem," in *Advances in Artificial Intelligence*, vol. 5549 of *Lecture Notes in Computer Science*, pp. 229–232, Springer, Berlin, Germany, 2009.

[43] W. S. Havens and B. N. Dilkina, "A hybrid schema for systematic local search," in *Advances in Artificial Intelligence: 17th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2004, London, Ontario, Canada, May 17–19, 2004. Proceedings*, vol. 3060 of *Lecture Notes in Computer Science*, pp. 248–260, Springer, Berlin, Germany, 2004.

[44] N. Jussien and O. Lhomme, "Local search with constraint propagation and conflict-based heuristics," *Artificial Intelligence*, vol. 139, no. 1, pp. 21–45, 2002.

[45] P. V. Hentenryck and L. Michel, *Constraint-Based Local Search*, MIT Press, 2005.

[46] B. W. Kerninghan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.

[47] S. Lin and B. W. Kerninghan, "An effective heuristic for the traveling salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[48] U. Grenander, *Pattern Analysis*, Springer, Berlin, Germany, 1978.

[49] J. Bentley, "Programming pearls: perspective on performance," *Communications of the ACM*, vol. 27, no. 11, pp. 1087–1092, 1984.

[50] C. Lecoutre, 2010, https://www.cril.univ-artois.fr/~lecoutre/benchmarks.html.

[51] K. Xu, F. Boussemart, F. Hemery, and C. Lecoutre, "Random constraint satisfaction: easy generation of hard (satisfiable) instances," *Artificial Intelligence*, vol. 171, no. 8-9, pp. 514–534, 2007.

[52] K. Xu and W. Li, "Exact phase Transition in constraint satisfaction problems," *Journal of Artificial Intelligence Research*, vol. 12, pp. 93–103, 2000.

[53] C. Solnon, *Ant Colony Optimization and Constraint Programming*, Wiley-ISTE, 2006.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 238529, 10 pages
http://dx.doi.org/10.1155/2015/238529

Research Article

Moving Clusters within a Memetic Algorithm for Graph Partitioning

Inwook Hwang,[1] Yong-Hyuk Kim,[2] and Yourim Yoon[3]

[1]*Technical Laboratory, Atto Research, 225-18 Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do 463-400, Republic of Korea*
[2]*Department of Computer Science & Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 139-701, Republic of Korea*
[3]*Department of Computer Engineering, College of Information Technology, Gachon University, 1342 Seongnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do 461-701, Republic of Korea*

Correspondence should be addressed to Yourim Yoon; yryoon@gachon.ac.kr

Received 23 September 2014; Accepted 7 January 2015

Academic Editor: John Gunnar Carlsson

Most memetic algorithms (MAs) for graph partitioning reduce the cut size of partitions using iterative improvement. But this local process considers one vertex at a time and fails to move clusters between subsets when the movement of any single vertex increases cut size, even though moving the whole cluster would reduce it. A new heuristic identifies clusters from the population of locally optimized random partitions that must anyway be created to seed the MA, and as the MA runs it makes beneficial cluster moves. Results on standard benchmark graphs show significant reductions in cut size, in some cases improving on the best result in the literature.

1. Introduction

Consider an unweighted undirected graph $G = (V, E)$, where V is a set of n vertices, and E is the set of edges that connect them. A *k-way partition* $\{P_1, P_2, \ldots, P_k\}$ of the graph G is a partitioning of the vertex set V into k disjoint subsets. A partition is said to be *balanced* if the difference in size between the largest and the smallest subset is at most 1, that is, for all $1 \leq i, j \leq k$, $||P_i| - |P_j|| \leq 1$. The *cut size* of a partition is defined to be the number of edges connecting vertices in different subsets of the partition. The *k-way graph partitioning* problem is the problem of finding a balanced k-way partition with the minimum cut size. If $k = 2$, it can be called bipartitioning and if $k > 2$, multiway partitioning. These problems arise in applications such as sparse matrix factorization, network partitioning, layout and floor planning, circuit placement, social network analysis, and software-defined networking [1, 2].

For general graphs, partitioning is known to be NP-hard [3]. Bui and Jones [4] have shown that even finding good approximate solutions is also NP-hard.

Therefore, many heuristic methods have been proposed: some of them work well, but they cannot of course guarantee optimality. The simplest heuristic is iterative improvement partitioning (IIP) [5, 6], exemplified by the Kernighan-Lin (KL) [7] and the Fiduccia-Mattheyses (FM) algorithms [8], but these algorithms only produce solutions which are approximations to *local* optima; however, this limitation can be overcome by hybridizing them with metaheuristics, such as simulated annealing [9], genetic algorithms (GAs) [10], tabu search [11, 12], or ant colony optimization [13]. Recently, a number of techniques based on GAs have achieved notable results for $k = 2$ [14–19] and $k > 2$ [20–26]. Kim et al. [27] have surveyed this work.

The use of IIP for local optimization of partitioning produced by a GA becomes less effective as the graph becomes larger. We will show that this is because IIP often fails to move densely interconnected subgraphs, called *clusters*, between partitions, and hence fails to find partitions with small cut sizes.

The goal of the work reported in this paper is to overcome the barriers to effective search which are presented

by clusters, by modifying the GA so that it contributes to move clusters appropriately. We present a memetic algorithm (MA), which is a GA combined with local optimization, in which a heuristic finds clusters in some of the positions in each generation, by examining population of individuals, each of which represents a position, rather than trying to identify them directly from a single graph. It moves some of these clusters. This heuristic supplements the well-known ability of MAs to provide attractive initial points for local optimization. Experimental results show that this approach can substantially improve the performance of an MA. The contributions of this work are summarized as follows.

(i) We provide a detailed explanation of the difficulty of moving clusters in graph partitioning and provide experimental results quantifying the impact of clusters on the search for partitions with a small cut size.

(ii) We present a heuristic for detecting and moving clusters, which is based on a new, population-based, measure of the distance between vertices called *genic distance*.

(iii) We show that this heuristic substantially improves the ability of an MA to find good partitions.

The remainder of this paper is organized as follows. In Section 2 we briefly introduce IIP algorithms and the test graphs used in our experiments. In Section 3 we investigate the difficulty of moving clusters in graph partitioning. In Section 4 we describe our new cluster-handling heuristic, and an MA that uses this heuristic is described in Section 5. In Section 6 we present experimental results, and draw conclusions in Section 7.

2. Preliminaries

2.1. Iterative Improvement Algorithms in Bipartitioning. Iterative improvement partitioning starts with a random partition. This is refined in a series of passes. At the start of each pass, all the vertices are free to move between subsets. IIP selects vertices and moves them, but each vertex is only moved once during a pass. At the end of the pass, the best partition found during the pass is identified and used as the input to the next pass. Passes continue until there is no further improvement.

There are a number of IIP algorithms, of which KL [7] is often considered to be the first reasonable heuristic for bipartitioning. In KL, the movement of vertices during a pass is restricted to the swapping of a pair of vertices between subsets.

Let $\{A, B\}$ be a partition of V into two subsets A and B. We define the gain g_v associated with a vertex v to be the reduction in cut size obtained by moving v to the other subset. By extension, the gain $g(a, b)$ obtained by swapping vertices $a \in A$ and $b \in B$ can be expressed as follows:

$$g(a,b) = g_a + g_b - 2\delta(a,b), \tag{1}$$

where

$$\delta(a,b) = \begin{cases} 1, & \text{if } (a,b) \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

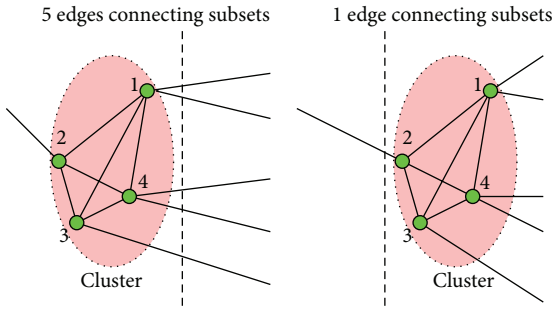
KL selects the pair (a, b) with the highest value of $g(a, b)$ and effects the exchange. The vertices a and b are not considered again during the current pass. A sequence of pairs $(a_1, b_1), (a_2, b_2), \ldots, (a_{n/2-1}, b_{n/2-1})$ are selected in this way. The algorithm chooses l that maximizes $\sum_{i=1}^{l} g(a_i, b_i)$ and exchanges $\{a_1, a_2, \ldots, a_l\}$ and $\{b_1, b_2, \ldots, b_l\}$. KL performs further passes until no improvement is possible.

FM is another widely used IIP algorithm, which is similar to KL, except that it only moves one vertex at a time. This makes FM faster than KL, with little loss in partition quality. Several variants of KL and FM exist [15, 28, 29].

2.2. Local Optimization Algorithms for Multiway Partitioning. There are three main schemes for multiway partitioning, which are developments of the recursive, pair-wise, and direct approaches [21] to bipartitioning. The recursive KL algorithm bisects the graph recursively until there are k subsets. The pair-wise KL [7] starts with an arbitrary k-way partition. It picks two subsets at a time from the k subsets and performs bipartitioning to reduce the cut size between those pairs. Sanchis [30] extended the FM algorithm to multiway partitioning and showed that the direct method performed better than recursion. The extended algorithm considers moving each vertex from its current subset to every other subset. To perform local optimization in the proposed MA for multiway partitioning, we use a variant of this algorithm, called EFM (extended FM) [21]. The time complexity of EFM is $O(k|E|)$.

2.3. Local Search in Memetic Algorithms. It is already clear that combining a GA with local optimization algorithms is an effective approach to the graph partitioning problem [15]. Some authors have explored fast but weak local optimization algorithms. For example [31, 32], 2-opt was used to relocate border vertices, which are those with edges that connect to vertices in other subsets. Bui and Moon [10] obtained better results with KL by allowing only a single pass, while restricting the number of vertices to be swapped. Conversely, other authors have reported notable improvements by enhancing local optimization algorithms. For bipartitioning, Kim and Moon [15] suggested a new KL-based local optimization algorithm, formulated using a new type of gain, called *lock gain*, which only takes into account the edges that connect a vertex to the vertices that have already been moved. Combined with a GA, this algorithm obtained impressive results on most benchmark graphs. For multiway partitioning, the combination of MAs with specialized local optimization algorithms showed good results [20, 21]. Steenbeek et al. [18] proposed what they called a cluster enhancement heuristic, which they combined with an MA, and reported successful results. Their MA uses a vertex swap heuristic to identify clusters. The MA only handles the moving of clusters between subsets.

2.4. Test Graphs. We tested our MA on Johnson's benchmark graphs [9], which have been widely used in other studies [10, 11, 14–17, 20, 21, 23, 33–36]. They are composed of 8 random



(a) Before moving a cluster (b) After moving a cluster

Figure 1: An example of cluster moving, in which the cut size of a partition is reduced by 4.

graphs G$n.d$ and 8 random geometric graphs U$n.d$. The two different classes of graphs are briefly described below.

(i) G$n.d$: a random graph on n vertices, with an independent probability p that any two vertices are connected by an edge. The probability p is biased so that the expected vertex degree, $p(n-1)$, is d.

(ii) U$n.d$: a random geometric graph on n vertices that lie in the unit square and whose coordinates are chosen uniformly from the unit interval. Every pair of vertices separated by a distance of t or less is connected by an edge. The expected degree of a vertex is $n\pi t^2$.

3. Difficulty of Moving Clusters

Suppose that the cluster shown in Figure 1(a) is involved in a bipartitioning problem. The four vertices in this cluster are fully interconnected, and they all belong to the same subset. Moving this cluster to the other subset, across the dotted line in Figure 1(b) will reduce the cut size of the partition by 4. However, there is no motivation to move any single vertex, because they all have negative gain: the gains of v_1, v_2, v_3, and v_4 are −1, −4, −2, and −1, respectively. This example illustrates how IIP algorithms may miss a significant reduction in cut size that could be achieved by moving several vertices together.

The baleful effect of clusters on local search algorithms trying to solve the graph partitioning problem motivated this study. Kim [37, 38] indicated that graph partitioning is hard primarily due to the difficulty of moving clusters. Dutt and Deng [39, 40] have also observed that an IIP method applied to circuit partitioning can fail because of the difficulty of dealing with clusters that straddle subsets.

3.1. Experimental Support. We designed experiments to quantify the effect of clusters on IIP algorithms, represented by the KL algorithm. Using the cluster detection method to be described in Section 4.1, we find clusters in the graph and select one randomly. We then take a locally optimum bipartition s obtained by KL and move the selected cluster to the other subset, creating a perturbed partition t_{cluster}. Applying KL to t_{cluster}, we obtain a new local optimum u_{cluster}.

Table 1: Probability that KL fails to return vertices moved from one subset of a partition to the other, when the vertices are in cluster (P_{cluster}) or chosen at random (P_{random}), over 1,000 runs.

Graph	P_{cluster} (%)	P_{random} (%)	Average number of vertices moved (q)
G500.2.5	16.90	2.50	6.41
G500.05	5.70	1.30	5.50
G500.10	1.90	0.30	5.05
G500.20	0.50	0.10	4.78
G1000.2.5	12.80	1.90	5.89
G1000.05	2.90	0.80	5.61
G1000.10	0.60	0.20	4.67
G1000.20	0.30	0.00	4.91
U500.05	39.10	1.00	7.41
U500.10	26.20	0.50	12.05
U500.20	8.60	0.20	14.10
U500.40	10.40	0.40	49.36
U1000.05	37.90	0.80	7.81
U1000.10	28.50	0.60	12.47
U1000.20	11.60	0.10	22.19
U1000.40	9.10	0.40	35.88

Assuming that q, the number of vertices in the clusters, is small, u_{cluster} can be expected to match s if KL is successful in moving the cluster back. However, if KL fails to return the perturbed cluster to its original subset, the cut size of the partition may increase. Repeating this experiment, we derive P_{cluster}, as an approximation of the probability that the cut size of u_{cluster} is larger than that of s.

For comparison, we perturbed q vertices randomly selected within a locally optimized partition, by moving them to the other subset. We call this partition t_{random}. We apply the KL algorithm to t_{random} and then obtain a new locally optimum partition u_{random}. Repeating this experiment, we derive P_{random}, as an approximation of the probability that the cut size of u_{random} is larger than that of s.

Table 1 shows the values of P_{cluster} and P_{random} for 16 benchmark graphs. We see that P_{cluster} is always larger than P_{random}, as we would expect. We notice that the gap between P_{cluster} and P_{random} is larger on the geometric graphs (U$n.d$) than on the random graphs (G$n.d$).

4. Cluster-Handling Heuristic

Cluster analysis is a well-known problem for which plenty of algorithms exist, many of which require a lot of computation. The insight that motivates our heuristic is that the application of a local optimization process, such as IIP, to a randomly partitioned graph creates a modified partition in which clusters tend to be wholly allocated to one subset or another (and are then difficult to move, as we have already observed). A single partition of this sort is of little help in identifying clusters, because the clusters are not separated at all within each subset; but if we create many random partitions and optimize

Table 2: Truth table for Fact 1.

p	q	$p \rightarrow q$	$I(p)$	$I(q)$
True	True	True	1	1
True	False	False	1	0
False	True	True	0	1
False	False	True	0	0

Table 3: Truth table for Fact 2.

p	q	$I(p)$	$I(q)$	$p \vee q$	$I(p \vee q)$	$I(p) + I(q)$
True	True	1	1	True	1	2
True	False	1	0	True	1	1
False	True	0	1	True	1	1
False	False	0	0	False	0	0

them, we can reasonably infer that vertices that find themselves in the same subset in most of these partitions belong to the same cluster. We can make this inference in a structured way using the "genic distance" metric that we will introduce. This approach to cluster analysis may seem indirect, but it is efficient in the context of an evolutionary approach to graph partitioning, because the set of partitions required for finding clusters using genic distance is also the population which we must create to be evolved by our MA.

One way of dealing with clusters is to devise a local optimization heuristic that can identify clusters [18, 19, 38]. However, this prevents us from building on previous studies of IIP algorithms.

Our approach is to add an additional heuristic to our MA, which finds and moves clusters. The heuristic identifies clusters in the population of partitions which have already been optimized locally. It selects clusters with higher gains and moves them. IIP local optimization is then applied again.

4.1. Cluster Detection. Let $I(\cdot)$ be an *indicator function*, that is, $I(\text{true}) = 1$ and $I(\text{false}) = 0$. Then, we can trivially establish the following.

Fact 1. If $p \rightarrow q$ is true, then $I(p) \leq I(q)$.

Proof. From Table 2. □

Fact 2. $I(p \vee q) \leq I(p) + I(q)$.

Proof. From Table 3. □

We now define a metric called *genic distance*, which measure the extent to which two vertices connected by an edge can be considered to belong to the same cluster. We denote the genic distance of an edge $\{v_i, v_j\}$ within a population $\{p_1, p_2, \ldots, p_m\}$ of locally optimized partitions as $d_g(v_i, v_j)$, which can be expressed as follows:

$$d_g\left(v_i, v_j\right) := \sum_{l=1}^{m} I\left(p_l\left(g_i\right) \neq p_l\left(g_j\right)\right), \tag{3}$$

where g_i and g_j are the genes corresponding to v_i and v_j, respectively. The value of gene g_i (i.e., the partition number that vertex v_i belongs to) in the lth individual is $p_l(g_i)$. For convenience, we assume that each vertex has an edge that connects it to itself, so that $\{v, v\} \in E$ for each vertex v. Then, the following proposition holds.

Proposition 1. *For each population, d_g becomes a pseudometric on V.*

Proof. Since $I(\cdot) \geq 0$, $d_g(v_i, v_j) = \sum_{l=1}^{m} I(p_l(g_i) \neq p_l(g_j)) \geq 0$ for each $\{v_i, v_j\} \in E$. It is enough to show the following three conditions.

(i) $d_g(v_i, v_i) = 0$:

$$\begin{aligned} d_g\left(v_i, v_i\right) &= \sum_{l=1}^{m} I\left(p_l\left(g_i\right) \neq p_l\left(g_i\right)\right) \\ &= \sum_{l=1}^{m} I\left(\text{false}\right) \\ &= 0. \end{aligned} \tag{4}$$

(ii) *Symmetry*. Let $\{v_i, v_j\}$ be in E:

$$\begin{aligned} d_g\left(v_i, v_j\right) &= \sum_{l=1}^{m} I\left(p_l\left(g_i\right) \neq p_l\left(g_j\right)\right) \\ &= \sum_{l=1}^{m} I\left(p_l\left(g_j\right) \neq p_l\left(g_i\right)\right) \\ &= d_g\left(v_j, v_i\right). \end{aligned} \tag{5}$$

(iii) *Triangle Inequality*. Consider each group of three edges $\{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$. If $p_l(g_i) = p_l(g_k)$ and $p_l(g_k) = p_l(g_j)$, then $p_l(g_i) = p_l(g_j)$ for each l. By contraposition, if $p_l(g_i) \neq p_l(g_j)$, then $p_l(g_i) \neq p_l(g_k)$ or $p_l(g_k) \neq p_l(g_j)$.

For each l,

$$\begin{aligned} &I\left(p_l\left(g_i\right) \neq p_l\left(g_j\right)\right) \\ &\leq I\left(p_l\left(g_i\right) \neq p_l\left(g_k\right) \vee p_l\left(g_k\right) \neq p_l\left(g_j\right)\right) \quad (\because \text{Fact 1}) \\ &\leq I\left(p_l\left(g_i\right) \neq p_l\left(g_k\right)\right) + I\left(p_l\left(g_k\right) \neq p_l\left(g_j\right)\right) \quad (\because \text{Fact 2}). \end{aligned} \tag{6}$$

By summing the above inequalities for all l,

$$\begin{aligned} &\sum_{l=1}^{m} I\left(p_l\left(g_i\right) \neq p_l\left(g_j\right)\right) \\ &\leq \sum_{l=1}^{m} \left(I\left(p_l\left(g_i\right) \neq p_l\left(g_k\right)\right) + I\left(p_l\left(g_k\right) \neq p_l\left(g_j\right)\right)\right) \\ &= \sum_{l=1}^{m} I\left(p_l\left(g_i\right) \neq p_l\left(g_k\right)\right) + \sum_{l=1}^{m} I\left(p_l\left(g_k\right) \neq p_l\left(g_j\right)\right). \end{aligned} \tag{7}$$

Therefore we have

$$d_g(v_i, v_j) \le d_g(v_i, v_k) + d_g(v_k, v_j). \quad (8)$$

That is, d_g satisfies the triangle inequality. □

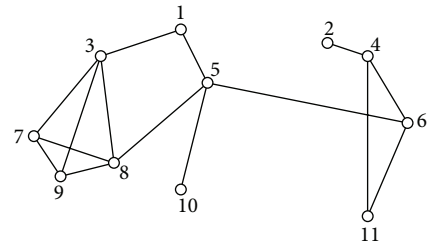
Proposition 1 suggests that the measure d_g is reasonable. A pseudometric space is a generalization of a metric space, in which points need not be distinguishable; thus it is possible that $d_g(v_i, v_j) = 0$ for some edge $\{v_i, v_j\}$, with distinct vertices $v_i \neq v_j$.

Our heuristic detects clusters by collecting a number of local optima and computes genic distances for all the edges in each graph. This takes $O(m|E|)$ time, but this cost is negligible since this computation is a preprocess performed before the MA runs. The heuristic temporarily eliminates edges with genic distances that are greater than a threshold value θ. We set θ to be the smallest value that satisfies

$$\left|\left\{e \in E : d_g(e) \le \theta\right\}\right| \le 0.1\,|V|. \quad (9)$$

Each remaining connected component containing more than three vertices is considered to be a *cluster*.

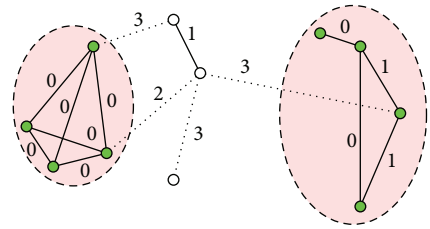
Figure 2 shows how our heuristic detects clusters. Figure 2(a) shows an example graph with 11 vertices and 15 edges. Four individuals, corresponding to locally optimized partitions, from the population are shown in Figure 2(b). In Figure 2(c), each edge is labeled with its genic distance. If the threshold value of genic distance is 1, then the edges with larger genic distances, indicated by dotted line, are eliminated. Then four connected components remain: $\{v_3, v_7, v_8, v_9\}$, $\{v_2, v_4, v_6, v_{11}\}$, $\{v_1, v_5\}$, and $\{v_{10}\}$. The last two of these connected components are considered too small to be clusters. Thus clusters $\{v_3, v_7, v_8, v_9\}$ and $\{v_2, v_4, v_6, v_{11}\}$, shaded in Figure 2(c), remain as candidates for moving.



(a) Example graph

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

(b) Four locally optimized partitions in the population (subsets 0 and 1)



(c) Two detected clusters

Figure 2: An example of cluster detection.

4.2. Cluster-Moving Scheme. Our heuristic improves the offspring of each generation after crossover by moving the clusters that were detected using the technique described in the previous subsection. To select the clusters to be moved and their target subsets, we introduce a measure called *cluster gain*, such that $\mathrm{cg}(x, a)$ is the reduction in the cut size of the partition when all the vertices in cluster x are moved to the subset a. For example, moving the cluster in Figure 1(a) to the other subset in the partition is associated with a cluster gain of 2.

This cluster-moving scheme, described in Algorithm 1, is applied to each individual generated by crossover, which is a partition that may be unbalanced. However, cut size and cluster gain are well defined on unbalanced partitions. Our scheme does not consider moving every cluster in every partition, because we found that making all clusters movable causes the premature convergence of the MA. Thus, our heuristic selects N clusters at random as candidates for moving. In our experiments, we set N to 5. We compute the cluster gain that results from moving each candidate cluster to each of the other $k - 1$ subsets. The candidate cluster x and destination subset a with the highest cluster gain are selected. Assume that $\mathrm{cg}(x, a)$ is positive, all the vertices in cluster x are moved to subset a, and cluster x is removed from the set of candidate clusters. This process is repeated until no candidates remain, or no move yields a positive cluster gain. No attempt is made to balance the partition during cluster-moving; this is performed later.

5. Memetic Search Framework

CMPA (cluster-moving memetic partitioning algorithm) is a memetic algorithm that we have designed for graph partitioning. In this MA, an individual is a k-way partition. Each gene in an individual corresponds to a vertex and has a value between 0 and $k - 1$, which indicates the subset to which the vertex belongs; that is, the ith gene $g_i = j \Leftrightarrow v_i \in P_{j+1}$. It is a steady-state MA, meaning that there is only one offspring from population in each generation. Crossover is followed by a cluster-moving step and then local optimization.

Algorithm 2 shows the processes that make up CMPA, which we now describe in detail.

(i) *Initialization*. When the MA starts, m individuals (i.e., partitions) are created at random. Then each individual is improved by local optimization. We set m to be 30 for bipartitioning and 50 for multiway partitioning (with $k = 8$ or $k = 32$).

```
Select clusters: S ← N clusters selected at random;
do {
   Calculate cg(x, a) for all x ∈ S, 1 ≤ a ≤ k;
   cg* ← max_{x,a} cg(x, a);
   (x̄, ā) ← arg max_{x,a} cg(x, a);
   if cg* > 0 then {
      Move cluster x̄ to partition ā;
      S ← S \ {x̄};
   }
} until (S = ∅ or cg* ≤ 0)
```

Algorithm 1: Our cluster-moving scheme.

```
Create initial population of fixed size;
Apply local optimization to each member of population;
Calculate genic distance from population;
Find clusters and store their information;
do {
   Select parent_1 and parent_2 from population;
   Normalization(parent_1, parent_2);
   offspring ← Crossover(parent_1, parent_2);
   Cluster-moving(offspring);
   Local-optimization(offspring);
   Replace(population, offspring);
} until (stopping condition);
return the best solution;
```

Algorithm 2: The process in CMPA.

(ii) *Selection*. We used the roulette-wheel-based proportional selection. The probability that the best individual is chosen was set to four times the probability that the worst is chosen. The fitness value f_i of the ith individual is expressed as $(c_w - c_i) + (c_w - c_b)/3$, where c_b, c_w, and c_i are the cut sizes of the partitions corresponding to the best, the worst, and the ith individual, respectively.

(iii) *Normalization*. Laszewski [31] first used normalization to improve the performance of GA and its variants have been suggested in [23, 26, 41, 42]. The parent individuals are normalized before crossover following Laszewski [31, 33]. The subset of one parent which shares the largest number of vertices with subsets of the other parent is selected, and that subset is numbered 0. This process is repeated, incrementing the index, until all subsets are numbered.

(iv) *Crossover and Cluster Moving*. We used a standard five-point crossover. After crossover, the cluster-handling heuristic described in Section 4.2 is applied to the individual. At this point, individuals usually correspond to unbalanced partitions. We select a random location in the individual and adjust the values of the genes, which are the subsets to which the corresponding vertices belong, to the right of this location (in a typical circular string) until the partition is balanced. This is effectively a mutation effect, and no further mutation was introduced.

(v) *Local Optimization*. KL [7] was used for the bisection problems and EFM (extended FM) [21] was used for multiway partitioning ($k = 8$ or $k = 32$).

(vi) *Replacement*. We used a replacement scheme due to Bui and Moon [10]. If an offspring is better than its closer parent, the MA replaces that parent. If it is better than its other parent, then that parent is replaced. Otherwise it replaces the worst individual in the population.

(vii) *Stopping Condition*. This is based on consecutive failures to replace an individual's parents. Termination is triggered by consecutive failures: 30 in bipartitioning and 50 in multiway partitioning.

6. Experimental Results

We conducted experiments on 2-way, 8-way, and 32-way partitioning. Table 4 shows the performance of MA combined with KL (denoted KL-MA) and CMPA on bipartitioning. Table 5 shows the performance of the genetic extended FM algorithm (GEFM) [21], one of the most effective approaches, and CMPA on 8-way partitioning, and Table 6 gives the results for 32-way partitioning. CMPA uses a cluster-handling

Table 4: Comparison of KL-MA and CMPA on bipartitioning.

Graph	Best known[1]	KL-MA			CMPA		
		Best[2]	Ave[2]	CPU[3] (Gen[4])	Best[2]	Ave[2]	CPU[3] (Gen[4])
G500.2.5	49	49	51.34	0.32 (595)	49	**51.22**	0.34 (558)
G500.05	218	218	218.65	0.59 (700)	218	**218.45**	0.64 (682)
G500.10	626	626	627.55	1.02 (681)	626	**627.46**	1.10 (672)
G500.20	1744	1744	1745.83	1.93 (694)	1744	**1745.62**	2.09 (687)
G1000.2.5	93	93	97.26	0.96 (745)	93	**97.08**	1.04 (722)
G1000.05	445	445	451.50	2.07 (884)	445	**450.92**	2.22 (860)
G1000.10	1362	1362	1366.68	4.07 (969)	1362	**1366.18**	4.45 (977)
G1000.20	3382	3382	3385.61	7.42 (971)	3382	**3385.26**	8.13 (987)
U500.05	2	2	4.96	0.55 (793)	2	**3.94**	0.66 (803)
U500.10	26	26	26.31	0.45 (459)	26	**26.04**	0.57 (465)
U500.20	178	178	178.06	0.42 (300)	178	**178.00**	0.53 (271)
U500.40	412	412	**412.00**	0.34 (190)	412	**412.00**	0.37 (102)
U1000.05	1	1	11.67	1.71 (1077)	1	**8.81**	2.12 (1068)
U1000.10	39	39	46.27	1.51 (645)	39	**44.15**	1.95 (653)
U1000.20	222	222	222.19	1.13 (347)	222	**222.07**	1.51 (327)
U1000.40	737	737	**737.00**	1.20 (243)	737	**737.00**	1.77 (205)

[1]Best result from the literature.
[2]Best and average results from 1,000 runs.
[3]CPU seconds on a Pentium IV 2.8 GHz.
[4]Average number of generations over 1,000 runs.

Table 5: Comparison of GEFM and CMPA on 8-way partitioning.

Graph	GEFM [21]			CMPA		
	Best[1]	Ave[1]	CPU[2] (Gen[3])	Best[1]	Ave[1]	CPU[2] (Gen[3])
G500.2.5	111	115.21	42.03 (2090)	111	**114.51**	40.25 (2091)
G500.05	465	468.16	85.65 (2314)	465	**467.63**	72.96 (2239)
G500.10	1254	1259.53	144.54 (2347)	1254	**1258.47**	135.04 (2297)
G500.20	3348	3354.80	293.18 (2437)	3348	**3353.71**	285.09 (2317)
G1000.2.5	212	**216.30**	146.76 (3043)	211	216.57	148.42 (3023)
G1000.05	930	**938.11**	247.20 (3148)	931	939.26	234.67 (3069)
G1000.10	2714	2726.33	408.96 (3146)	2711	**2725.52**	416.98 (3083)
G1000.20	6525	**6536.55**	791.29 (3303)	6520	6538.35	781.67 (3141)
U500.05	16	18.80	70.87 (2629)	16	**17.22**	55.89 (2137)
U500.10	143	145.12	98.75 (2277)	143	**144.26**	90.50 (1984)
U500.20	612	614.57	121.22 (1690)	611	**612.93**	97.53 (1483)
U500.40	1867	1872.60	157.94 (1396)	1867	**1871.73**	129.62 (1204)
U1000.05	21	33.66	180.35 (3638)	20	**28.53**	209.53 (3172)
U1000.10	176	183.70	243.08 (3105)	176	**182.30**	257.02 (2857)
U1000.20	812	814.05	231.08 (1824)	812	**813.00**	232.93 (1756)
U1000.40	2562	2563.05	249.46 (1353)	2562	**2562.85**	273.51 (1238)

[1]Best and average results from 100 runs.
[2]CPU seconds on a Pentium IV 2.8 GHz.
[3]Average number of generations over 100 runs.

heuristic but KL-MA and GEFM do not. We performed 1,000 runs for bipartitioning and 100 runs for 8-way and 32-way partitioning. All the programs were written in the C language and compiled using GNU's *gcc* compiler. It was run on a Pentium IV 2.8 GHz computer with Linux. In the tables, the bold-faced numbers indicate the lower of the average cut sizes obtained by the two algorithms. CMPA outperformed KL-MA and GEFM on most graphs, with comparable running times.

CMPA underperformed on some random graphs, which may be due to the weak cluster structures in these graphs. CMPA's average performance was better on all the geometric

Table 6: Comparison of GEFM and CMPA on 32-way partitioning.

Graph	Best known[1]	GEFM [21]			CMPA		
		Best[2]	Ave[2]	CPU[3] (Gen[4])	Best[2]	Ave[2]	CPU[3] (Gen[4])
G500.2.5	177	177	181.69	78.98 (1491)	177	**180.82**	74.30 (1382)
G500.05	623	624	630.47	162.26 (2197)	623	**629.92**	159.18 (2035)
G500.10	1573	1574	1581.94	300.83 (2364)	1572*	**1579.72**	355.73 (2332)
G500.20	4034	4037	4045.06	681.53 (2565)	4035	**4042.81**	673.19 (2479)
G1000.2.5	312	313	320.99	335.79 (2798)	313	**320.25**	356.44 (2648)
G1000.05	1217	1208	1218.72	698.67 (3133)	1205*	**1216.89**	738.06 (3203)
G1000.10	3360	3353*	3367.31	1171.48 (3294)	3355	**3366.08**	1323.59 (3267)
G1000.20	7818	7817	**7829.81**	2012.78 (3475)	7815*	7830.48	2363.69 (3342)
U500.05	109	112	116.39	138.29 (1640)	109	**113.18**	121.15 (1595)
U500.10	523	531	537.75	225.52 (1512)	526	**531.84**	166.26 (1222)
U500.20	1825	1831	1841.39	285.07 (1344)	1823*	**1831.88**	299.47 (1197)
U500.40	5328	5364	5380.01	561.74 (1381)	5348	**5369.83**	523.15 (1331)
U1000.05	117	118	126.02	451.03 (3137)	115*	**123.49**	464.35 (2967)
U1000.10	577	576	583.26	599.01 (2599)	571*	**578.16**	743.50 (2603)
U1000.20	2367	2375	2396.33	798.77 (1963)	2373	**2388.93**	930.43 (1848)
U1000.40	7329	7399	7417.49	1421.78 (1634)	7382	**7407.18**	1493.10 (1573)

[1]Best known values [20–22, 24, 33].
[2]Best and average results from 100 runs. Asterisked numbers are new best values.
[3]CPU seconds on a Pentium IV 2.8 GHz.
[4]Average number of generations from 100 runs.

graphs. This suggests that effective cluster handling is more important on the geometric graphs, as we suggested in Section 3.1.

The local optimization algorithm is much more expensive than the cluster-handling heuristic; thus, CMPA does not take much longer to run than KL-MA or GEFM. On average, CMPA required 14.14% more time than KL-MA for the bipartitioning problems, and 2.02% more than GEFM in 32-way partitioning. CMPA ran 5.52% faster in 8-way partitioning.

7. Concluding Remarks

We devised a cluster-moving heuristic and combined it with a memetic algorithm (MA) for graph partitioning. Experiments on 2-way, 8-way, and 32-way partitioning showed that this heuristic significantly improved the performance of the MA, especially on the 32-way partitioning.

The method of moving clusters that we have introduced addresses a significant weakness in standard IIP algorithms. The idea of adding an operation to complement a local optimization algorithm might be used to address other deficiencies in MAs.

Our method of cluster detection is based on a measure that we call genic distance, which is designed to reflect the extent to which two vertices connected by an edge belong to the same cluster. Instead of computing genic distances once in an initialization phase, an MA could recompute these distances as it progresses: this might improve the accuracy of cluster detection, at some cost in time. We believe that genic distance might also be useful in solving other problems involving clusters, such as the maximum clique problem.

Table 7: Real-world benchmark graphs.

Graph	Number of vertices	Number of edges
nasa4704	4704	50026
bcspwr09	1723	2394
bcsstk13	2003	40940
DEBR12	4096	8189

Appendix

Results on Real-World Graphs

We also tested on four real-world benchmark graphs used in [11, 15, 43]. The sizes of the graphs are given in Table 7. We conducted experiments on 2-way and 8-way partitioning. We performed 100 runs for bipartitioning and 50 runs for 8-way partitioning. It was run on an Intel Core i7 3.6 GHz computer with Linux. Table 8 compares CMPA with KL-MA on bipartitioning, and Table 9 compares CMPA with GEFM on 8-way partitioning. In the tables, the bold-faced numbers indicate the lower of the average cut sizes obtained by the two algorithms. Similarly to the results in Section 6, CMPA overall performed better than the others, with comparable running times.

Disclosure

A preliminary version of this paper appeared in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2007 (p. 1520).

Table 8: Comparison of KL-MA and CMPA on bipartitioning.

Graph	Best known[1]	KL-MA			CMPA		
		Best[2]	Ave[2]	CPU[3] (Gen[4])	Best[2]	Ave[2]	CPU[3] (Gen[4])
nasa4704	1292	1292	**1292.00**	4.15 (407)	1292	**1292.00**	3.82 (347)
bcspwr09	9	9	11.31	0.61 (1010)	9	**10.60**	0.81 (1003)
bcsstk13	2355	2355	**2355.00**	1.17 (330)	2355	**2355.00**	1.46 (282)
DEBR12	548	548	548.44	9.91 (765)	548	**548.20**	10.32 (794)

[1]Best result from the literature.
[2]Best and average results from 100 runs.
[3]CPU seconds on Intel Core i7 3.6 GHz.
[4]Average number of generations over 100 runs.

Table 9: Comparison of GEFM and CMPA on 8-way partitioning.

Graph	GEFM [21]			CMPA		
	Best[1]	Ave[1]	CPU[2] (Gen[3])	Best[1]	Ave[1]	CPU[2] (Gen[3])
nasa4704	3898	3903.66	575.69 (2235)	3896	**3902.66**	591.51 (2255)
bcspwr09	53	**57.08**	78.80 (2947)	54	57.74	85.16 (2925)
bcsstk13	8911	8939.10	281.19 (1427)	8919	**8936.14**	301.54 (1476)
DEBR12	1248	1260.16	315.97 (3969)	1248	**1259.90**	353.83 (4002)

[1]Best and average results from 50 runs.
[2]CPU seconds on Intel Core i7 3.6 GHz.
[3]Average number of generations over 50 runs.

Conflict of Interests

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

Acknowledgments

This work was supported by the ICT R&D program of MSIP/IITP, Korea (10045253, The development of SDN/OpenFlow-Based Enterprise Network Controller Technology) project. The authors would like to thank Professor Byung-Ro Moon and Dr. Yongjoo Song for their valuable suggestions for improving this paper. The authors also thank Jong-Pil Kim for providing the source code of GEFM [21].

References

[1] Y. Kanizo, D. Hay, and I. Keslassy, "Palette: distributing tables in software-defined networks," in *Proceedings of the IEEE International Conference on Computer Communications*, pp. 545–549, April 2013.

[2] P. Wette, M. Draxler, and A. Schwabe, "MaxiNet: distributed emulation of software-defined networks," in *Proceedings of the IFIP Networking Conference*, pp. 1–9, June 2014.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[4] T. N. Bui and C. Jones, "Finding good approximate vertex and edge partitions is NP-hard," *Information Processing Letters*, vol. 42, no. 3, pp. 153–159, 1992.

[5] Y.-H. Kim, "Improved implementation choices for iterative improvement partitioning algorithms on circuits," in *Proceedings of the International Conference on Computer Design*, pp. 30–34, July 2008.

[6] Y. Yoon and Y.-H. Kim, "New bucket managements in iterative improvement partitioning algorithms," *Applied Mathematics & Information Sciences*, vol. 7, no. 2, pp. 529–532, 2013.

[7] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.

[8] C. Fiduccia and R. Mattheyses, "A Linear-time heuristic for improving network partitions," in *Proceedings of the Conference on Design Automation*, pp. 175–181, June 1982.

[9] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing. An experimental evaluation. Part I. Graph partitioning," *Operations Research*, vol. 37, no. 6, pp. 865–892, 1989.

[10] T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," *IEEE Transactions on Computers*, vol. 45, no. 7, pp. 841–855, 1996.

[11] R. Battiti and A. A. Bertossi, "Greedy, prohibition, and reactive heuristics for graph partitioning," *IEEE Transactions on Computers*, vol. 48, no. 4, pp. 361–385, 1999.

[12] E. Rolland, H. Pirkul, and F. Glover, "Tabu search for graph partitioning," *Annals of Operations Research*, vol. 63, no. 2, pp. 209–232, 1996.

[13] T. N. Bui and L. C. Strite, "An ant system algorithm for graph bisection," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 43–51, July 2002.

[14] I. Hwang, Y.-H. Kim, and B.-R. Moon, "Multi-attractor gene reordering for graph bisection," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1209–1215, July 2006.

[15] Y.-H. Kim and B.-R. Moon, "Lock gain based graph partitioning," *Journal of Heuristics*, vol. 10, no. 1, pp. 37–57, 2004.

[16] P. Merz and B. Freisleben, "Fitness landscapes, memetic algorithms, and greedy operators for graph bipartitioning," *Evolutionary Computation*, vol. 8, no. 1, pp. 61–91, 2000.

[17] H. Mühlenbein and T. Mahnig, "Evolutionary optimization and the estimation of search distributions with applications to graph

bipartitioning," *International Journal of Approximate Reasoning*, vol. 31, no. 3, pp. 157–192, 2002.

[18] A. G. Steenbeek, E. Marchiori, and A. E. Eiben, "Finding balanced graph bi-partitions using a hybrid genetic algorithm," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 90–95, May 1998.

[19] Y. Yoon and Y.-H. Kim, "Vertex ordering, clustering, and their application to graph partitioning," *Applied Mathematics & Information Sciences*, vol. 8, no. 1, pp. 135–138, 2014.

[20] S.-J. Kang and B.-R. Moon, "A hybrid genetic algorithm for multiway graph partitioning," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 159–166, July 2000.

[21] J.-P. Kim and B.-R. Moon, "A hybrid genetic search for multiway graph partitioning based on direct partitioning," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 408–415, July 2001.

[22] Y.-H. Kim, A. Moraglio, Y. Yoon, and B.-R. Moon, "Geometric crossover for multiway graph partitioning," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1217–1224, July 2006.

[23] A. Moraglio, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "Geometric crossovers for multiway graph partitioning," *Evolutionary Computation*, vol. 15, no. 4, pp. 445–474, 2007.

[24] A. Moraglio, Y.-H. Kim, Y. Yoon, B.-R. Moon, and R. Poli, "Geometric crossover for permutations with repetitions: application to graph partitioning," in *Proceedings of the PPSN Workshop on Evolutionary Algorithms—Bridging Theory and Practice*, September 2006.

[25] A. J. Soper, C. Walshaw, and M. Cross, "A combined evolutionary search and multilevel optimisation approach to graph-partitioning," *Journal of Global Optimization*, vol. 29, no. 2, pp. 225–241, 2004.

[26] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "Quotient geometric crossovers and redundant encodings," *Theoretical Computer Science*, vol. 425, pp. 4–16, 2012.

[27] J. Kim, I. Hwang, Y.-H. Kim, and B.-R. Moon, "Genetic approaches for graph partitioning: a survey," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 473–480, July 2011.

[28] S. Dutt and W. Deng, "A probability-based approach to VLSI circuit partitioning," in *Proceedings of the Design Automation Conference*, pp. 100–105, June 1996.

[29] B. Krishnamurthy, "An improved min-cut algorithm for partitioning VLSI networks," *IEEE Transactions on Computers*, vol. 33, no. 5, pp. 438–446, 1984.

[30] L. A. Sanchis, "Multiple-way network partitioning," *IEEE Transactions on Computers*, vol. 38, no. 1, pp. 62–81, 1989.

[31] G. Laszewski, "Intelligent structural operators for the k-way graph partitioning problem," in *Proceedings of the International Conference on Genetic Algorithms*, pp. 45–52, July 1991.

[32] H. Mühlenbein, "Parallel genetic algorithms in combinatorial optimization," in *Computer Science and Operations Research*, O. Balci, R. Sharda, and S. Zenios, Eds., pp. 441–453, Pergamon, Oxford, UK, 1992.

[33] S.-S. Choi and B.-R. Moon, "Normalization in genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2723 of *Lecture Notes in Computer Science*, pp. 862–873, 2003.

[34] S.-H. Kim, Y.-H. Kim, and B.-R. Moon, "A hybrid genetic algorithm for the max cut problem," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 416–423, July 2001.

[35] Y.-H. Kim, Y.-K. Kwon, and B.-R. Moon, "Problem-independent schema synthesis for genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1112–1122, July 2003.

[36] K. Seo, S. Hyun, and Y.-H. Kim, "An edge-set representation based on spanning tree for searching cut space," *IEEE Transactions on Evolutionary Computation*, 2014.

[37] Y.-H. Kim, *Combinatorial optimization based on effective local search, genetic algorithms, and problem space analyses [Ph.D. thesis]*, Seoul National University, Seoul, Republic of Korea, 2005.

[38] Y.-H. Kim, "An enzyme-inspired approach to surmount barriers in graph bisection," in *Proceedings of the International Conference on Computational Science and Its Applications*, vol. 5072 of *Lecture Notes in Computer Science*, pp. 841–851, 2008.

[39] S. Dutt and W. Deng, "VLSI circuit partitioning by cluster-removal using iterative improvement techniques," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 194–200, November 1996.

[40] S. Dutt and W. Deng, "Cluster-aware iterative improvement techniques for partitioning large VLSI circuits," *ACM Transactions on Design Automation of Electronic Systems*, vol. 7, no. 1, pp. 91–121, 2002.

[41] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1473–1483, 2013.

[42] Y. Yoon, Y.-H. Kim, A. Moraglio, and B.-R. Moon, "A theoretical and empirical study on unbiased boundary-extended crossover for real-valued representation," *Information Sciences*, vol. 183, no. 1, pp. 48–65, 2012.

[43] B. Monien and R. Diekmann, "A local graph partitioning heuristic meeting bisection bounds," in *Proceedings of the SIAM Conference on Parallel Processing for Scientific Computing*, March 1997.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 869627, 7 pages
http://dx.doi.org/10.1155/2015/869627

Research Article

A Real-Time Pothole Detection Approach for Intelligent Transportation System

Hsiu-Wen Wang,[1] Chi-Hua Chen,[2] Ding-Yuan Cheng,[3] Chun-Hao Lin,[2] and Chi-Chun Lo[1]

[1]*Institute of Information Management, National Chiao Tung University, Hsinchu 300, Taiwan*
[2]*Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taoyuan 326, Taiwan*
[3]*Department of Information Management, Hwa Hsia University of Technology, New Taipei 235, Taiwan*

Correspondence should be addressed to Chi-Hua Chen; chihua0826@gmail.com

Received 14 August 2014; Revised 4 December 2014; Accepted 9 December 2014

Academic Editor: Jung-Fa Tsai

In recent years, fast economic growth and rapid technology advance have led to significant impact on the quality of traditional transport system. Intelligent transportation system (ITS), which aims to improve the transport system, has become more and more popular. Furthermore, improving the safety of traffic is an important issue of ITS, and the pothole on the road causes serious harm to drivers' safety. Therefore, drivers' safety may be improved with the establishment of real-time pothole detection system for sharing the pothole information. Moreover, using the mobile device to detect potholes has been more popular in recent years. This approach can detect potholes with lower cost in a comprehensive environment. This study proposes a pothole detection method based on the mobile sensing. The accelerometer data is normalized by Euler angle computation and is adopted in the pothole detection algorithm to obtain the pothole information. Moreover, the spatial interpolation method is used to reduce the location errors from global positioning system (GPS) data. In experiments, the results show that the proposed approach can precisely detect potholes without false-positives, and the higher accuracy is performed by the proposed approach. Therefore, the proposed real-time pothole detection approach can be used to improve the safety of traffic for ITS.

1. Introduction

In recent years, fast economic growth and rapid technology advance have led to significant impact on the quality of traditional transport system. Intelligent transportation system (ITS), which aims to improve the transport system, has become more and more popular. For the safety of traffic, road users often feel uncomfortable when they drive on the rough roads, especially the potholes on the road. In accordance with the statistics from the Ministry of Justice in Taiwan, the national compensation money is about 240 million dollars from 2008 to 2011 [1]. The pothole on the road causes serious harm to drivers' safety. Therefore, drivers' safety may be improved with the establishment of real-time pothole detection system for sharing the pothole information.

Moreover, more and more sensors which include *G*-sensors, electronic compass, gyroscope, global positioning system (GPS), microphone, and cameras are equipped in mobile device (e.g., smartphone and iPad). Several applications use these sensors in mobile devices and combine mobile sensing techniques to solve problems such as social network [2], healthcare [3], environment monitoring [4], and traffic information [5]. Therefore, using the mobile device based on mobile sensing techniques to detect potholes is suitable and convenient.

This study proposes a pothole detection method based on the mobile sensing and shares the pothole information with road users and government. For this purpose, the mobile device should be equipped with *G*-sensors and GPS to collect accelerometer data and location information. The accelerometer data is normalized by Euler angle computation and is adopted in the pothole detection algorithm to obtain the pothole information. Moreover, the spatial interpolation method is used to reduce the location errors from GPS data. Then the pothole information is made public to improve the safety of traffic.

The remainder of the paper is organized as follows. Section 2 presents and discusses the various techniques of image recognition method and mobile sensing method for pothole detection. In Section 3, this study proposes a real-time pothole detection method based on mobile sensing. This study also illustrates the experiment results and analyses in Section 4. Finally, this study concludes the paper in Section 5.

2. Related Work

Some pothole detection methods have been proposed and can be classified into two groups: image recognition method and mobile sensing method. The advantages and shortcomings of these methods are presented in the following subsections.

2.1. Image Recognition Method. Yu and Salari proposed a pothole detection approach based on laser imaging techniques to collect road information. Then the artificial neural network algorithm (ANN) was used to analyze the road information and detect potholes [6]. However, this approach which requires a big computation power to recognize the laser images is unsuitable for mobile devices.

Lin and Liu used the support vector machine algorithm (SVM) to analyze images about road information for pothole detection [7]. Although this approach can provide high accuracy, a big computation power is required for image recognition. Therefore, this approach is also unsuitable for mobile devices.

2.2. Mobile Sensing Method. For BusNet project, the *G*-sensor and GPS are equipped in the on-board unit (OBU) in bus to collect accelerometer data and location information. These data can be sent to data processing center via wireless networks, and data processing center can analyze these data to check whether the vectors of accelerometer data exceed the thresholds for pothole detection [8]. However, this approach requires that the batch accelerometer data is sent when bus enters the bus station. Therefore, this approach cannot provide real-time pothole detection information.

The pothole patrol system which was proposed by a project team from Massachusetts Institute of Technology combined *G*-sensor and GPS. This system analyzed the x-axis accelerometer data and z-axis accelerometer data and equipped five data filters which include (1) speed, (2) high-pass, (3) z-peak, (4) xz-ratio, and (5) speed versus z ratio [9]. Although these data filters can detect potholes, only z-peak of data filter can obtain the precise pothole information. However, high misjudgment of z-peak of data filters with the surge of road.

Nericell project used a smartphone based on Windows Mobile operation system which is equipped with *G*-sensor and GPS to collect and analyze accelerometer data for pothole detection [10]. However, the smartphone in this project should be equipped with the specific angle. Furthermore, this project only considered analyzing z-axis accelerometer data with high misjudgment.

Mednis et al. proposed four pothole detection approaches which include (1) *Z*-THRESH approach, (2) *Z*-DIFF approach, (3) STDEV-*Z* approach, and (4) *G*-ZERO approach to analyze the accelerometer data [11]. The accelerometer data in this study is obtained from Tmote sensors, Texas Instruments controllers, and Analog Devices *G*-sensors [12]. However, the results of *Z*-THRESH approach and *G*-ZERO approach would be influenced by peak value to generate more false-positives. The results of *Z*-DIFF approach and STDEV-*Z* approach are dependent on frequency and timing. Therefore, the design and comparisons of these approaches for mobile device require to be investigated.

2.3. Summary. In summary, due to the requirement of big computation power for image recognition method, too many resources are allocated for this method to be an efficiency mobile device. Therefore, mobile sensing method is more suitable to detect potholes for mobile device. However, previous pothole detection approaches based on mobile sensing only considered one threshold to detect pothole, and high false-positives are obtained from these approaches. Consequently, this study considers Mednis's approaches [11] and proposes a real-time pothole detection method to improve the accuracy of pothole detection method.

3. Real-Time Pothole Detection Method

The proposed real-time pothole detection method based on mobile sensing includes three steps: (1) accelerometer data normalization, (2) pothole detection approaches, and (3) pothole location determination.

3.1. Problem Definition. Some shortcomings are in previous pothole detection methods as follows: (1) mobile device should be equipped with the specific angle; (2) high false-positives may be generated with considering only one threshold for pothole detection; (3) the precise pothole location has not been investigated.

Therefore, this study proposes a real-time pothole detection method based on mobile sensing to collect and normalize the accelerometer data from mobile device for free angle establishment. Furthermore, a pothole detection algorithm is proposed to consider several thresholds and combine several pothole detection approaches for pothole detection accuracy improvement. Finally, the space interpolation method is adopted to determine pothole location for leaving shortcomings.

3.2. Accelerometer Data Normalization. For solving the limitation of the specific angle in previous pothole detection approaches, this study uses Euler angle formulas to normalize the three-axis accelerometer data. The Euler angles describe the vector set in 3-dimensional Euclidean space three parameters and represent a sequence of three elemental rotations. For example, Figure 1 shows that the vector set of accelerometer data is defined as $\{x', y', z'\}$. The $x'y'z'$ system rotates about the x'-axis by angle α. The y'-axis is now at angle α with respect to the y-axis, and the z'-axis is now at angle α with respect to the z-axis. In accordance with

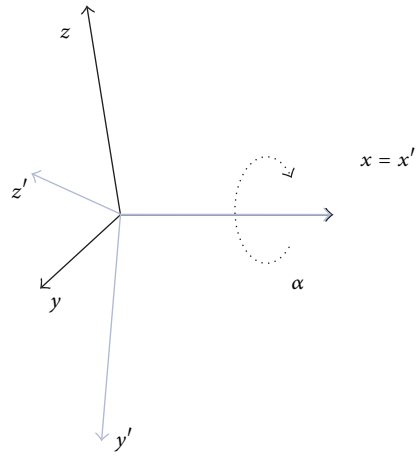


Figure 1: A case study of Euler angles (the $x'y'z'$ system rotates about the x'-axis by angle α).

Euler angle formulas, the vector set $\{x, y, z\}$ can be calculated by adopting the values of vector set $\{x', y', z'\}$ and angle α (shown in formulas (1)). Furthermore, the vector of each axis can be calculated by using Euler angle formulas when the system rotates about the y'-axis by angle β and z'-axis by angle γ (shown in formulas (2)). Therefore, the vector of each axis with 0 degree angle is referred to as baseline in this study. In runtime stage, the vector set $\{x, y, z\}$ can be calculated by adopting the vector set of baseline and rotation angle for accelerometer data normalization:

$$
\begin{aligned}
&[x \;\; y \;\; z] = [x' \;\; y' \;\; z'] \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \\
&x = x', \\
&y = y' \times (\cos\alpha) + z' \times (\sin\alpha), \\
&z = y' \times (-\sin\alpha) + z' \times (\cos\alpha),
\end{aligned} \tag{1}
$$

$$
\begin{aligned}
&[x \;\; y \;\; z] = [x' \;\; y' \;\; z'] \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, \\
&x = x' \times (\cos\beta) + z' \times (-\sin\beta), \\
&y = y', \\
&z = x' \times (\sin\beta) + z' \times (\cos\beta),
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
&[x \;\; y \;\; z] = [x' \;\; y' \;\; z'] \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
&x = x' \times (\cos\gamma) + y' \times (\sin\gamma), \\
&y = x' \times (-\sin\gamma) + y' \times (\cos\gamma), \\
&z = z'.
\end{aligned}
$$

3.3. Pothole Detection Approaches. This study considers the previous four pothole detection approaches and proposes a pothole detection algorithm to combine these approaches for pothole detection improvement. The notations of these approaches are defined and summarized in Notations.

3.3.1. The First Pothole Detection Approach: Z-THRESH. The *Z*-THRESH approach considers the minimum value of z-axis accelerometer data as the threshold to detect pothole. The value of z-axis accelerometer data is about $-9.80065\ \mathrm{m/s^2}$ when the *G*-sensor is laid horizontally. Moreover, the value of z-axis accelerometer data is lower than $-9.80065\ \mathrm{m/s^2}$ when the *G*-sensor is dropped off. Therefore, the z-axis accelerometer data drops off quickly when the car enters a pothole. Then z-axis accelerometer data increases when the car leaves a pothole. Therefore, this study considers the lowest value of z-axis accelerometer data through a pothole in experimental runs. Furthermore, the maximum value of the minimum value of z-axis accelerometer data in each run is selected as the threshold θ_1 which is suitable to detect pothole for each experimental run (shown in formula (3)). In runtime stage, the value of $f_1(g_{a,i,j})$ is 1 when the value of $g_{a,i,j}$ is lower than θ_1 for pothole detection (shown in formula (4)).

The value of threshold is

$$
\theta_1 = \max_{a=1,1\le i\le n, i\in N} \; \min_{e_i\le j\le l_i, j\in N} g_{a,i,j}. \tag{3}
$$

Detection function is

$$
f_1\left(g_{a,i,j}\right) = \begin{cases} 1, & \text{if } g_{a,i,j} \le \theta_1 \\ 0, & \text{others,} \end{cases} \tag{4}
$$

$$
\text{where } a = 1, \quad 1 \le i \le n, \quad i \in N, \quad j \in N.
$$

3.3.2. The Second Pothole Detection Approach: Z-DIFF. The *Z*-DIFF approach considers the maximum difference of two consecutive z-axis accelerometer records as the threshold to detect pothole. Due to much decreasing and increasing of z-axis accelerometer data through a pothole, the velocity of variation of z-axis accelerometer data between time $t_{i,j-1}$ and time $t_{i,j}$ is calculated and used to detect pothole. Therefore, this study retrieves the largest value of velocity of variation of z-axis accelerometer data through a pothole in experimental runs. Furthermore, the minimum value of the maximum value of variation velocity in each run is selected as the threshold θ_2 which is suitable to detect pothole for each experimental run (shown in formula (5)). In runtime stage, the value of $f_2(g_{a,i,j})$ is 1 when the value of $|g_{a,i,j} - g_{a,i,j-1}|/(t_{i,j} - t_{i,j-1})$ is larger than θ_2 for pothole detection (shown in formula (6)). However, the limitation of this approach is difficult to determine the time difference between $t_{i,j-1}$ and $t_{i,j}$, and the accuracy of this approach is influenced by this time difference.

The value of threshold is

$$
\theta_2 = \min_{a=1,1\le i\le n, i\in N} \; \max_{e_i\le j\le l_i, j\in N} \frac{\left|g_{a,i,j} - g_{a,i,j-1}\right|}{t_{i,j} - t_{i,j-1}}. \tag{5}
$$

Detection function is

$$f_2\left(g_{a,i,j}\right) = \begin{cases} 1, & \text{if } \dfrac{\left|g_{a,i,j} - g_{a,i,j-1}\right|}{t_{i,j} - t_{i,j-1}} \geq \theta_2 \\ 0, & \text{others,} \end{cases} \tag{6}$$

where $a = 1, \quad 1 \leq i \leq n, \quad i \in N, \quad j \geq 1, \quad j \in N.$

3.3.3. The Third Pothole Detection Approach: STDEV(Z). The STDEV(Z) approach considers the maximum standard deviation of z-axis accelerometer data as the threshold to detect pothole. Due to the perturbation motion of z-axis accelerometer data through a pothole, the standard deviation of z-axis accelerometer data during K records is calculated and used to detect pothole. Therefore, this study gets the maximum value of standard deviation of z-axis accelerometer data through a pothole in experimental runs. Furthermore, the minimum value of the maximum value of standard deviation in each run is selected as the threshold θ_3 which is suitable to detect pothole for each experimental run (shown in formula (7)). In runtime stage, the value of $f_3(g_{a,i,j})$ is 1 when the value of $\sqrt{\sum_{k=j-K+1}^{j}(g_{a,i,k} - \mu_{i,j})^2/K}$ is larger than θ_3 for pothole detection (shown in formula (8)). However, the limitation of this approach is difficult to determine the value of K which means time period, and the accuracy of this approach is influenced by this time period.

The value of threshold is

$$\theta_3 = \min_{a=1,1\leq i\leq n,i\in N} \max_{e_i\leq j\leq l_i,j\in N} \sqrt{\frac{\sum_{k=j-K+1}^{j}\left(g_{a,i,k} - \mu_{i,j}\right)^2}{K}}, \tag{7}$$

$$\text{where } \mu_{i,j} = \frac{\sum_{k=j-K+1}^{j} g_{a,i,k}}{K}.$$

Detection function is

$$f_3\left(g_{a,i,j}\right) = \begin{cases} 1, & \text{if } \sqrt{\dfrac{\sum_{k=j-K+1}^{j}\left(g_{a,i,k} - \mu_{i,j}\right)^2}{K}} \geq \theta_3 \\ 0, & \text{others,} \end{cases} \tag{8}$$

$$\text{where } a = 1, \quad 1 \leq i \leq n, \quad i \in N, \quad j \geq K,$$

$$j \in N, \quad \mu_{i,j} = \frac{\sum_{k=j-K+1}^{j} g_{a,i,k}}{K}.$$

3.3.4. The Fourth Pothole Detection Approach: G-ZERO. The G-ZERO approach considers all three-axis accelerometer data and selects a lower bound and upper bound to detect pothole for accuracy improvement. When a car passes through a pothole, all values of three-axis accelerometer data are near to zero. Therefore, this study considers the largest value of three-axis accelerometer data through a pothole as a candidate of lower bound and the lowest value of three-axis accelerometer data through a pothole as a candidate of upper bound. Then the minimum value of the maximum value of three-axis accelerometer data in each run is selected as the lower bound $\theta_{4,1}$, and the maximum value of the minimum value of three-axis accelerometer data in each run is selected as the upper bound $\theta_{4,2}$ (shown in formulas (9) and (10)). In runtime stage, the value of $f_4(g_{a,i,j})$ is 1 when the value of $g_{a,i,j}$ is larger than $\theta_{4,1}$ and is lower than $\theta_{4,2}$ for pothole detection (shown in formula (11)).

The value of lower bound is

$$\theta_{4,1} = \min_{a\in\{1,2,3\},1\leq i\leq n,i\in N} \max_{e_i\leq j\leq l_i,j\in N} g_{a,i,j}. \tag{9}$$

The value of upper bound is

$$\theta_{4,2} = \max_{a\in\{1,2,3\},1\leq i\leq n,i\in N} \min_{e_i\leq j\leq l_i,j\in N} g_{a,i,j}. \tag{10}$$

Detection function is

$$f_4\left(g_{a,i,j}\right) = \begin{cases} 1, & \text{if } \theta_{4,1} \leq g_{a,i,j} \leq \theta_{4,2} \\ 0, & \text{others,} \end{cases} \tag{11}$$

where $a = \{1, 2, 3\}, \quad 1 \leq i \leq n, \quad i \in N, \quad j \in N.$

3.3.5. The Fifth Pothole Detection Approach: Proposed Approach. This subsection proposes a pothole detection approach which combines and improves the Z-THRESH and G-ZERO approaches to detect pothole. Furthermore, the Z-DIFF and STDEV(Z) approaches are limited in accordance with time differences and time periods, so these two approaches are not adopted. The pseudocode of the proposed pothole detection approach is presented in Algorithm 1. The input parameters of this proposed approach are three-axis accelerometer data, and the value of output is 1 when the proposed pothole detection approach supposes the car passed through a pothole. In the proposed approach, the parameter *check_method* is used to record whether the value of $f_1(g_{a,i,j})$ or $f_4(g_{a,i,j})$ is 1. When one of Z-THRESH and G-ZERO approaches supposes that the car passed through a pothole, the timestamp $t_{i,j}$ is recorded and compared with the parameter *check_time*. The value of output is 1 if $t_{i,j}$-*check_time* is smaller than ε seconds, which means a pothole is detected. Furthermore, the parameter *check_time* can be trained and learned by historical data from each practical run.

3.4. Pothole Location Determination. For pothole location determination, this study uses the space interpolation method to obtain precise pothole location. Figure 2 shows that two locations (i.e., L_1 and L_2) and timestamps (i.e., t_1 and t_2) obtained from GPS module are adopted in the space interpolation method to determine the pothole location. The function $d(L_1, L_2)$ is defined as the distance between location L_1 and location L_2. Therefore, the pothole location L_p can be determined by using

$$d\left(L_1, L_p\right) = \frac{d\left(L_1, L_2\right) \times \left(t_2 - t_1\right)}{\left(t_3 - t_1\right)},$$

$$d\left(L_2, L_p\right) = \frac{d\left(L_1, L_2\right) \times \left(t_3 - t_2\right)}{\left(t_3 - t_1\right)}. \tag{12}$$

Input: $g_{a,i,j}$, where $a = \{1, 2, 3\}$, $1 \le i \le n, i \in N$, $j \in N$
Output: The value of output is 1 when the proposed pothole detection approach supposes the car passed through a pothole.
Set check_method = 0
set check_time = 0
while ($j \in N$)
  if ($t_{i,j}$-check_time) > ε sec **then**
    check_method = 0
    check_time = 0
  end if
  if $f_1(g_{1,i,j}) = 1$ **then**
    if check_method = 0 **then**
      check_method = 1
      check_time = $t_{i,j}$
    else if check_method = 4 **then**
      return 1
    else
      check_time = $t_{i,j}$
    end if
  end if
  if $f_4(g_{a,i,j}) = 1$ **then**
    if check_method = 0 **then**
      check_method = 4
      check_time = $t_{i,j}$
    else if check_method = 1 **then**
      return 1
    else
      check_time = $t_{i,j}$
    end if
  end if
end while

Algorithm 1: The pseudocode of the proposed pothole detection approach.

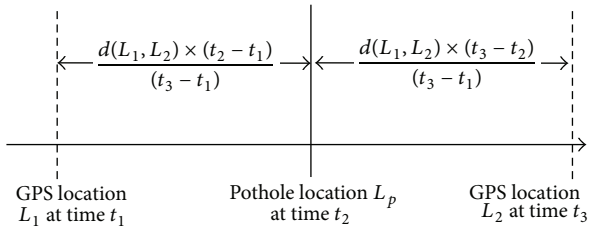


Figure 2: The space interpolation method for pothole location determination.

Figure 3: A case study of the mobile device with 0 degree angle as baseline.

4. Experimental Results

This section discusses the analyses of experimental results for accelerometer data normalization, pothole detection approaches, and pothole location determination.

4.1. The Analyses of Accelerometer Data Normalization. For the analyses of accelerometer data normalization, this study gives two case studies which include (1) the mobile device with 0 degree angle as baseline (shown in Figure 3) and (2) the mobile device with –15 degree angle (i.e., the value of α is –15 in Figure 1) (shown in Figure 4). Then the t-test and F-test are used to verify the difference between the z-axis accelerometer data of baseline in Case 1 and the z-axis accelerometer data after accelerometer data normalization in Case 2.

This study uses two-tailed t-test to determine significance of the difference between the mean of z-axis accelerometer data of baseline in Case 1 ($\mu_1 = -9.8489$) and the mean of z-axis accelerometer data after accelerometer data normalization in Case 2 ($\mu_2 = -9.8476$). The sample sizes of Case 1 and Case 2 are 60. Furthermore, this study also uses F-test to

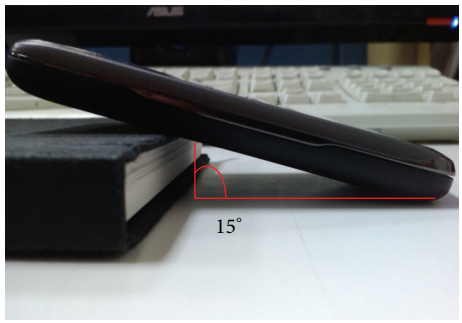


Figure 4: A case study of the mobile device with −15 degree angle.

determine significance of the difference between the variance of z-axis accelerometer data of baseline in Case 1 (σ_1^2 = 0.000077) and the variance of z-axis accelerometer data after accelerometer data normalization in Case 2 (σ_2^2 = 0.000085). Table 1 shows the t-test and F-test for the population means and variances of samples in Case 1 and Case 2. Experimental results indicate that the *null hypothesis* (H0: $\mu_1 = \mu_2$) in t-test is accepted, and another *null hypothesis* (H0: $\sigma_1 = \sigma_2$) in F-test is also accepted. Therefore, using Euler angle formulas to normalize the accelerometer data from mobile device is suitable for free angle establishment.

4.2. The Accuracy of Pothole Detection Approach. For the analyses of accuracy of pothole detection approach, this study selects a pothole (length: 58 cm; weight: 51 cm; and depth: 6 cm) as a case study (shown in Figure 5) and 10 runs in experiment environment. The mean frequency of accelerometer data detection in G-sensor in mobile device is 124 counts/second. The k-fold cross-validation [13] is used to verify the accuracy of pothole detection approach. In experiments, training and testing are performed 10 times (i.e., k = 10). In iteration i, the accelerometer data in ith run therapy is selected as the test corpus, and the accelerometer data in other runs is collectively used to train the thresholds for each approach. Table 2 shows the comparisons of Z-THRESH approach, Z-DIFF approach, STDEV(Z) approach, G-ZERO approach, and the proposed approach. The results show that the proposed approach can precisely detect potholes without false-positives and the accuracy of the proposed approach is 100%. Furthermore, this study also implemented and compared common machine learning methods which include ANN, SVM, and decision tree (DT) (shown in Table 3). Although these machine learning methods can detect potholes, several false-positives are generated by them.

4.3. The Error of Pothole Location Determination. For the analyses of error of pothole location determination, this study uses the accelerometer data and location information from 10 runs in Section 4.2 to verify the space interpolation method. The results show that the error of pothole location determination is reduced from 17.47 meters to 11.74 meters after using the space interpolation method. Therefore, the space interpolation method is suitable to determinate the precise pothole location.

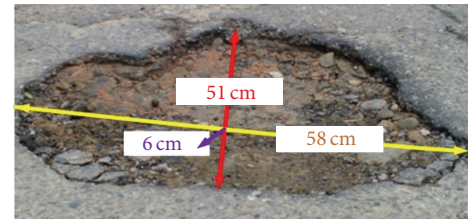


Figure 5: A case study of a pothole in University Road, Hsinchu, Taiwan.

Table 1: The test results for accelerometer data with normalization.

	The value of z-axis accelerometer data mean (standard deviation)
The 0 degree angle (baseline)	−9.8489 (0.000077)
The −15 degree angle after normalization	−9.8476 (0.000085)

Table 2: The false positive of each pothole detection approach.

Detection method	Z-THRESH	Z-DIFF	STDEV(Z)	G-ZERO	Proposed method
False-positive	49	40	274	8	0

Table 3: The comparisons of different machine learning methods.

Detection method	ANN	SVM	DT
False-positive	1626	255	282

5. Conclusions and Future Work

This study proposes a real-time pothole detection method based on the mobile sensing techniques. This method uses Euler angle computation to normalize the accelerometer data obtained from mobile device with free angle establishment. Moreover, a pothole detection approach is proposed to be combined with Z-THRESH and G-ZERO approaches for reducing the false-positives of pothole detection. Furthermore, the spatial interpolation method is adopted to obtain precisely the location of pothole. In experiments, the results show that the proposed approach can precisely detect potholes without false-positives and the accuracy of the proposed approach is 100%. Therefore, the proposed real-time pothole detection approach can be used to improve the safety of traffic for ITS.

However, the limitation of this study is sample size. In the future, more practical results will be retrieved and analyzed to deploy the proposed method everywhere. Furthermore, due to the limited battery capacity of mobile device, the issue about saving of computation power can be investigated. A green pothole detection approach is needed to reduce the frequency of accelerometer data detection with high accuracy of pothole detection.

Notations

$g_{a,i,j}$: The acceleration of the ath axle of the jth record in the ith run
$t_{i,j}$: The timestamp of the jth record in the ith run
n: The number of runs
e_i: The e_ith entering pothole record in the ith run
l_i: The l_ith leaving pothole record in the ith run
a: The coordinate axis of G-sensor in mobile device (e.g., the value of a is 1 which means Z-axis)
$f_m(\cdot)$: The output of the mth pothole detection approach (e.g., the value of $f_1(\cdot)$ is 1 when the first pothole detection approach supposes that the car passed through a pothole)
K: The third pothole detection approach requires K records to calculate the standard deviation
θ_m: The value of threshold for the mth approach
$\theta_{4,1}$: The value of lower bound for the fourth pothole detection approach
$\theta_{4,2}$: The value of upper bound for the fourth pothole detection approach.

Conflict of Interests

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

Acknowledgment

The research is supported by the National Science Council of Taiwan under Grants nos. NSC 102-2622-H-009-001-CC3, NSC102-2410-H146-002-MY2, NSC 102-2410-H-009-052-MY3, and MOST 103-2622-H-009-001-CC3.

References

[1] Ministry of Justice, National Compensation Statistics, and Statistics of Justice, 2012, http://www.moj.gov.tw/ct.asp?xItem=36988&CtNode=11625&mp=095.

[2] E. Miluzzo, N. D. Lane, K. Fodor et al., "Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)*, pp. 337–350, Raleigh, Calif, USA, November 2008.

[3] C.-C. Lo, C.-H. Chen, D.-Y. Cheng, and H.-Y. Kung, "Ubiquitous healthcare service system with context-awareness capability: design and implementation," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4416–4436, 2011.

[4] C. I. Wu, H. Y. Kung, C. H. Chen, and L. C. Kuo, "An intelligent slope disaster prediction and monitoring system based on WSN and ANP," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4554–4562, 2014.

[5] C.-H. Chen, H.-C. Chang, C.-Y. Su, C.-C. Lo, and H.-F. Lin, "Traffic speed estimation based on normal location updates and call arrivals from cellular networks," *Simulation Modelling Practice and Theory*, vol. 35, no. 1, pp. 26–33, 2013.

[6] X. Yu and E. Salari, "Pavement pothole detection and severity measurement using laser imaging," in *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT '11)*, pp. 1–5, Mankato, Minn, USA, May 2011.

[7] J. Lin and Y. Liu, "Potholes detection based on SVM in the pavement distress image," in *Proceedings of the 9th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES '10)*, pp. 544–547, Hong Kong, August 2010.

[8] K. de Zoysa, C. Keppitiyagama, G. P. Seneviratne, and W. W. A. T. Shihan, "A public transport system based sensor network for road surface condition monitoring," in *Proceedings of the Workshop on Networked Systems for Developing Regions*, Kyoto, Japan, August 2007.

[9] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*, pp. 29–39, June 2008.

[10] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, Raleigh, NC, USA, 2008.

[11] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, pp. 1–6, IEEE, Barcelona, Spain, June 2011.

[12] R. Zviedris, A. Elsts, G. Strazdins, A. Mednis, and L. Selavo, "LynxNet: wild animal monitoring using sensor networks," in *Proceedings of the 4th International Conference on Real-world Wireless Sensor Networks*, Colombo, Sri Lanka, 2010.

[13] J. Han and M. Kamber, *Data mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, Calif, USA, 2006.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 361901, 11 pages
http://dx.doi.org/10.1155/2015/361901

Research Article

Equilibrium Customer Strategies in the Queue with Threshold Policy and Setup Times

Peishu Chen,[1,2] Wenhui Zhou,[1] and Yongwu Zhou[1]

[1]*School of Business Administration, South China University of Technology, Guangzhou, Guangdong 510000, China*
[2]*Department of Mathematics, Chaohu University, Chaohu, Anhui 238000, China*

Correspondence should be addressed to Peishu Chen; cps8@sina.com

Received 10 October 2014; Revised 24 May 2015; Accepted 10 June 2015

Academic Editor: John Gunnar Carlsson

We consider the equilibrium behavior of customers in the $M/M/1$ queue with N policy and setup times. The server is shut down once the system is empty and begins an exponential setup time to start service again when the number of customers in the system accumulates the threshold N ($N \geq 1$). We consider the equilibrium threshold strategies for fully observable case and mixed strategies for fully unobservable case, respectively. We get various performance measures of the system and investigate some numerical examples of system size, social benefit, and expected cost function per unit time for the two different cases under equilibrium customer strategies.

1. Introduction

Due to wide applications for management in service system and modern electronic commerce, there exists an emerging trend to study the behavior of customers in queueing systems. In these models, the customers decide to join or balk in a decentralized way. Their decisions are affected by how much system information can be observed and how other customers make their decisions. This can be viewed as a game among the customers. Researches about the economic analysis of queueing models can go back at least to the pioneering work of Naor [1] who analyzed customers' optimal strategies in an observable Markovian queue with a concise reward-cost structure. Naor's model and results had already been extended by several authors; readers can refer to [2–4].

The concept of an N policy was first introduced by Yadin and Naor [5]. The server is immediately turned on whenever N ($N \geq 1$) or more customers are present in the system and turned off once there are no customers in the system. Once the server is shut down, the server may not operate until N customers are present in the model. Wang and Ke [6] studied the optimal operation of a single removable server in an $M/G/1$ queueing system with either infinite waiting room or finite waiting room by a supplementary variable technique, where the "removable server" is merely an abbreviation for the system of turning on and turning off the server depending on the number of customers in the queue. Wang et al. [7] further consider an $M/G/1$ queueing system with N policy, where the server is typically subject to breakdowns unpredictably.

Guo and Hassin [8] first consider customers' strategic behavior and social optimization in a single Markovian queue with N policy, in which the server is activated if there are N customers in the system and turned off once there are no customers in the system. They found an interesting conclusion which is that a customer can bring positive externality to others by joining, which means that a new arrival can either keep the server operational for a longer time or stir up an inactive server to come back to work sooner, which leads to promoting other customers' benefits. Guo and Li [9] further investigated customers' strategic behavior and social optimization in $M/M/1$ queues with N policy under two partial information cases.

Queues with setup times have also been investigated by many scholars [6, 10–12]. In queueing systems with setup times, once server is reactivated, a generally random time is required for setup before it can start serving customers. As for the research on the equilibrium customer behavior

in Markovian queues with setup times was first presented by Burnetas and Economou [13]. Subsequently, Economou and Kanta [14] further investigated the Markovian queue that alternates between on and off periods in observable cases. Chen and Zhou [15] studied three different scenarios of balking strategies in the Markovian queue with setup times and server breakdowns.

Frequent setups will inevitably lead to the operating cost being too high, so it is very crucial to determine when the server should start his service in many practical queueing systems. By utilizing N policy, the server does not turn on his service until the number of customers in the system accumulates a threshold N. Selecting appropriate value N can avoid excessively frequent setups. It is more practical to analyze equilibrium strategies for the N policy system with setup times. For example, from an economic perspective, a driver may start driving a shuttle whenever a certain amount of seats is occupied. Another example is a queue-based power-saving scheme to alleviate the energy hole problem [16, 17]. Handling with sensor nodes in wireless sensor networks (WSNs) is actually crucial to understand the optimal operation of the threshold policy for power saving in a WSN. In the "N policy" scheme, when the number of packets accumulates to N, the sensor node turns on its transmitting function of the radio sever and starts transmission process for the waiting packets. The sensor node is closed as soon as all present packets are transmitted. The N policy holds effective for a power-saving mechanism since it reduces the setup power consumption required to switch between a busy mode and an idle mode of a radio server in sensor nodes. The proposed queue-based power-saving technique can be used for prolonging the lifetime of the WSN effectively and economically. To the best of our knowledge, no other scientific paper has dealt with equilibrium customer strategies in the Markovian queue with N policy and setup times. Customers make their decisions based on a nature reward-cost structure, which incorporates their desire for service as well as their unwillingness to wait. We discuss, respectively, the fully observable scenario where customers are informed not only about the server state but also about the exact number of customers in the queue and the fully unobservable scenario where an arriving customer does not observe the exact number of customers waiting for service and the state of server at all.

The structure of this paper is organized as follows. In Section 2, we describe the dynamics of the queueing system. In Section 3, we obtain the equilibrium threshold strategies for the fully observable case and stationary probabilities of the corresponding queue. In Section 4, we derive the stationary probabilities and equilibrium mixed strategies for the fully unobservable case. In Section 5, some numerical results are presented to illustrate the effect of two different information levels and N policy on the customers' strategies. Finally, Section 6 provides some conclusions.

2. Model Description

We assume that there is a single server queue with infinite capacity where customers arrive according to a Poisson process with rate λ. The service times are supposed to be exponentially distributed with parameter μ. We further assume that $0 < \lambda < \mu$. The server utilizes the N policy: turn the server on as soon as N ($N \geq 1$) or more customers are present in the system and turn the server off whenever there are no customers in the system. After the server is shut down, the server cannot work until N customers are presented in the queue and a setup process begins for the server to be activated. The time needed to setup is exponentially distributed with rate θ. We further suppose that interarrival times, service times, and setup times are mutually independent.

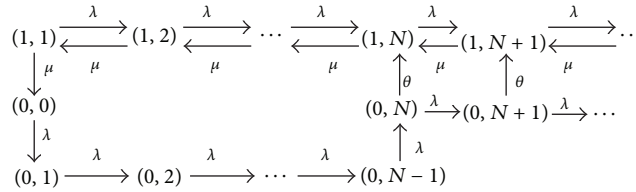


Figure 1: Transition rate diagram of the original model.

We denote the state of the system at time t by a random vector $(I(t), N(t))$, where $I(t)$ denotes the state of the server and $N(t)$ denotes the number of customers in the system. More specifically, $(0, n)$, $0 \leq n \leq N-1$, implies that the system is on N policy with n customers in the system; $(1, n)$, $n \geq N$, means that the system is on setup times with n customers in the system; $(2, n)$, $n \geq 1$, corresponds to the server which is busy with n customers in the system. It is clear that the process $\{(I(t), N(t)) : t \geq 0\}$ is a continuous time Markov chain with state space $S = \{(0,0), (0,1), \ldots, (0, N-1)\} \cup \{(1, N), (1, N+1), \ldots\} \cup \{(2,1), (2,2), \ldots\}$. The transition rate diagram is shown in Figure 1.

In this paper, we suppose that every customer gets a reward of R units for completing his service. Moreover, we suppose that there exists a waiting cost of C units per time unit that is continuously accumulated from the time that the customer joins the system till the time he leaves the system after he finished service.

3. Equilibrium Strategies and Some Performance Measures for the Fully Observable System

We now commit to the fully observable scenario, which assumes that customers observe not only the server state, but also the exact number of customers in the system. We will study the equilibrium threshold strategies and stationary probabilities for the fully observable case as well as some interesting performance measures of this system.

3.1. Equilibrium Threshold Strategies and Stationary Probabilities for the Fully Observable Case. In order to study the general threshold strategy adopted by all customers in the fully observable case, we will consider the mean overall sojourn time of the customer who is confronted with different server state upon arrival firstly. We start with the fully

observable case where customers know the exact state of the system and a customer in state (i, n) has mean sojourn time as follows.

(I) If a new customer finds the system in state $(0, n)$, where $0 \le n \le N-1$, then his expected benefit is

$$R - C\left(\frac{N-(n+1)}{\lambda} + \frac{1}{\theta} + \frac{n+1}{\mu}\right). \tag{1}$$

Let $f(x) = R - C((N-(x+1))/\lambda + 1/\theta + (x+1)/\mu)$; we have $f'(x) = C(1/\lambda - 1/\mu) > 0$ (since we assume that $0 < \lambda < \mu$ in Section 2). Obviously, $f(n)$ increases by increasing n. That is to say, when the system is on N policy, the least expected benefit is the first customer in the system. For the sake of ensuring a customer finds the system in $(0, 0)$ state and still joins the system, we must have

$$R - C\left(\frac{N-1}{\lambda} + \frac{1}{\theta} + \frac{1}{\mu}\right) \ge 0, \tag{2}$$

which means that

$$N \le \left\lfloor 1 + \frac{\lambda R}{C} - \frac{\lambda}{\theta} - \frac{\lambda}{\mu} \right\rfloor. \tag{3}$$

(II) If a new customer finds the system in state $(1, n)$, where $n \ge N$ (this phenomenon means the system is in setup phase), then his expected benefit is

$$R - C\left(\frac{1}{\theta} + \frac{n+1}{\mu}\right) \ge 0, \tag{4}$$

which means that if the server is currently in a setup state, the upper bound on the number of customers that an incoming customer is willing to stay behind is

$$n_e(1) = \left\lfloor \frac{R\mu}{C} - \frac{\mu}{\theta} \right\rfloor - 1. \tag{5}$$

(III) If a new customer finds the system in state $(2, n)$, then his expected benefit is

$$R - C\left(\frac{n+1}{\mu}\right) \ge 0, \tag{6}$$

which means that the upper bound on the number of service epochs that a customer is willing to wait is

$$n_e(2) = \left\lfloor \frac{R\mu}{C} \right\rfloor - 1. \tag{7}$$

Remark 1. Because any customer in N policy state must go through setup state and at last in service state, we have $N \le n_e(1) < n_e(2)$.

Theorem 2. *In the fully observable $M/M/1$ queue with setup times and N policy, a customer who observes the server in vacation is motivated to join if $N \le \lfloor 1 + \lambda R/C - \lambda/\theta - \lambda/\mu \rfloor$ or who observes the server in setup state is motivated to join if $n \le n_e(1)$ or who observes the server in service is motivated to join if $n \le n_e(2)$, or balk otherwise.*

Proof. From (3), we immediately find that a customer who observes the server in vacation is motivated to join if and only if $N \le \lfloor 1 + \lambda R/C - \lambda/\theta - \lambda/\mu \rfloor$. From (5), we find that if an arriving customer observes the server in setup state he is motivated to join on the condition that the number of customers in the system is no more than $n_e(1)$, which means that $n \le n_e(1)$. From (7), we find that if an arriving customer observes the server in service he is motivated to join if and only if $n \le n_e(2)$. If an arriving customer finds the number of customers in the system exceeds $n_e(2)$, which means that his expected net benefit is negative in any case, then his rational decision is to balk the system. □

We now turn our attention to the stationary distribution probabilities. Be aware that if all customers follow the threshold strategy in Theorem 2, then the system follows a Markov chain $(I(t), N(t))$ with state space of the system

$$\begin{aligned} S_{\text{ob}} = {} & \{(0,0), (0,1), \ldots, (0, N-1)\} \\ & \cup \{(1,n) \mid N \le n \le n_e(1) + 1\} \\ & \cup \{(2,n) \mid 1 \le n \le n_e(2) + 1\}. \end{aligned} \tag{8}$$

The transition diagram is depicted in Figure 2.

The corresponding stationary distribution $(p_{\text{ob}}(i, j) \in S_{\text{ob}})$ is given by the following theorem.

Theorem 3. *In the fully observable $M/M/1$ queue with N policy, setup times, and $\rho \ne \sigma$, in which the customers follow the $(N, n_e(1), n_e(2))$ threshold strategy, the stationary probabilities $(p_{ob}(i, j) \in S_{ob})$ are as follows:*

$$p_{ob}(0,n) = p_{ob}(0,0), \quad n = 1, 2, \ldots, N-1, \tag{9}$$

$$p_{ob}(1,n) = \sigma^{n-N+1} p_{ob}(0,0), \quad n = N, N+1, \ldots, n_e(1), \tag{10}$$

$$p_{ob}(1, n_e(1)+1) = \frac{\sigma^{n_e(1)-N+2}}{1-\sigma} p_{ob}(0,0), \tag{11}$$

$$p_{ob}(2,n) = \frac{\rho}{1-\rho}\left(1-\rho^n\right) p_{ob}(0,0), \quad n = 1, 2, \ldots, N, \tag{12}$$

$$\begin{aligned} & p_{ob}(2,n) \\ & = \frac{\rho\left\{(1-\rho)\sigma^{n+1-N} - \rho^n\left[\sigma - \rho + \rho^{1-N}(1-\sigma)\right]\right\}}{(1-\rho)(\sigma-\rho)} p_{ob}(0,0) \\ & \qquad n = N+1, N+2, \ldots, n_e(1)+1, \end{aligned} \tag{13}$$

$$\begin{aligned} p_{ob}(2,n) &= p_{ob}(2, n_e(1)+1)\,\rho^{n-n_e(1)-1}, \\ & \qquad n = n_e(1)+2, n_e(1)+3, \ldots, n_e(2)+1, \end{aligned} \tag{14}$$

$$\begin{aligned} p_{ob} = \Bigg\{ & N + \sum_{n=N}^{n_e(1)} \sigma^{n-N+1} + \frac{\sigma^{n_e(1)-N+2}}{1-\sigma} + \sum_{n=1}^{N} \frac{\rho(1-\rho^n)}{1-\rho} \\ & + \sum_{n=N+1}^{n_e(1)+1} \frac{\rho\left\{(1-\rho)\sigma^{n+1-N} - \rho^{n+1-N}\left[1-\sigma+\rho^{N-1}(\sigma-\rho)\right]\right\}}{(1-\rho)(\sigma-\rho)} \\ & + p_{ob}(1, n_e(1)+1) \sum_{n=n_e(1)+2}^{n_e(2)+1} \rho^{n-n_e(1)-1} \Bigg\}^{-1}, \end{aligned} \tag{15}$$

where $\rho = \lambda/\mu$ and $\sigma = \lambda/(\lambda+\theta)$.

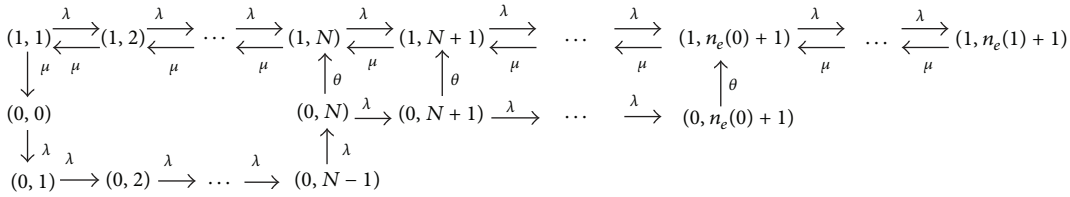


Figure 2: Transition rate diagram for the $(N, n_e(1), n_e(2))$ threshold strategy.

Proof. The stationary distribution of the system is gained from the following balance equations:

$$\lambda p(0,0) = \mu p(2,1), \tag{16}$$

$$\lambda p(0,n) = \lambda p(0,n-1), \quad n = 1,2,\ldots,N-1, \tag{17}$$

$$(\lambda+\theta)\, p(1,N) = \lambda p(0,N-1), \tag{18}$$

$$(\lambda+\theta)\, p(1,n) = \lambda p(1,n-1), \quad n = N, N+1, \ldots, n_e(1), \tag{19}$$

$$\lambda p\left(1, n_e(1)\right) = \theta p\left(1, n_e(1)+1\right), \tag{20}$$

$$\left(\lambda+\mu\right) p(2,1) = \mu p(2,2), \tag{21}$$

$$\left(\lambda+\mu\right) p(2,n) = \lambda p(2,n-1) + \mu p(2,n+1), \quad n = 2,3,\ldots,N-1, \tag{22}$$

$$\left(\lambda+\mu\right) p(2,n) = \theta p(1,n) + \lambda p(2,n-1) + \mu p(2,n+1), \quad n = N, N+1, \ldots, n_e(1)+1, \tag{23}$$

$$\left(\lambda+\mu\right) p(2,n) = \lambda p(2,n-1) + \mu p(2,n+1), \quad n = n_e(1)+2, n_e(1)+3, \ldots, n_e(2)+1, \tag{24}$$

$$\lambda p\left(2, n_e(2)\right) = \mu p\left(2, n_e(2)+1\right), \tag{25}$$

$$\sum_{n=0}^{N-1} p(0,n) + \sum_{n=N}^{n_e(1)+1} p(1,n) + \sum_{n=1}^{n_e(2)+1} p(2,n) = 1. \tag{26}$$

From (17), we easily get (9) in Theorem 3. From (18) and (19), we get (10) in Theorem 3. We further put $p_{\text{ob}}(1, n_e(1))$ in (20) and then we get exact expression of $p_{\text{ob}}(1, n_e(1)+1)$ in (11). From (16) and (21), we get $p_{\text{ob}}(2,1)$ and $p_{\text{ob}}(2,2)$ and put them into (22); we immediately get expression (12) in Theorem 3. We further put $p_{\text{ob}}(1, n_e(1))$ in (20); then we get exact expression of $p_{\text{ob}}(1, n_e(1)+1)$ in (11). From (16) and (21), we get $p_{\text{ob}}(2,1)$ and $p_{\text{ob}}(2,2)$ and then put them into (22); we immediately get expression (12) in Theorem 3. Next, we put $p_{\text{ob}}(2, N-1)$ and $p_{\text{ob}}(2, N)$ into expression (23); then we have (13) in Theorem 3. From (24) and (25), we have

$$\lambda p(2,n-1) = \mu p(2,n), \quad n = n_e(1)+2, n_e(1)+3, \ldots, n_e(2)+1. \tag{27}$$

By putting $p_{\text{ob}}(2, n_e(1)+1)$ into above recursive equation, we have (14). By using the normalizing equation (26), we get the expression of $p_{\text{ob}}(0,0)$ in Theorem 3 by putting (9)–(14) into (26). □

Remark 4. Theorem 3 holds for the stationary distribution corresponding to any threshold policy (N, n_1, n_2), $N \le n_1 \le n_2$, and not merely to the individually optimal policy specified by $n_1 = n_e(1)$ and $n_2 = n_e(2)$.

3.2. The System Performance Measures for the Fully Observable Case. We now study an expected cost function per unit time for the $(N, n_e(1), n_e(2))$ threshold strategy system. Our objectives are to analyze performance measures of the fully observable system and determine the optimum value of the important control threshold N by numerical computations.

(1) The Actual Arrival Rate. Because an arrival balks whenever he finds the system at state $(1, n_e(1)+1)$ or $(2, n_e(2)+1)$, then the system's effective arrival rate is given by

$$\lambda\left(n_e(1), n_e(2)\right) = \lambda\left[1 - p_{\text{ob}}\left(1, n_e(1)+1\right) - p_{\text{ob}}\left(2, n_e(2)+1\right)\right]. \tag{28}$$

(2) The Expected Number of Customers. Let L_{ob} denote the expected number of customers in the $(N, n_e(1), n_e(2))$ threshold strategy system. Then we obtain

$$L_{\text{ob}} = \frac{N(N-1)\, p_{\text{ob}}(0,0)}{2} + \sum_{n=N}^{n_e(1)+1} n p_{\text{ob}}(1,n) + \sum_{n=1}^{n_e(2)+1} n p_{\text{ob}}(2,n). \tag{29}$$

(3) The Probability of the Server in Different State. We further give the probability that the server in idle, setup, and busy steady state is as follows:

$$\begin{aligned} P(i=0) &= Np_{\text{ob}}(0,0); \\ P(i=1) &= \sum_{n=N}^{n_e(1)+1} p_{\text{ob}}(1,n); \\ P(i=2) &= \sum_{n=1}^{n_e(2)+1} p_{\text{ob}}(2,n). \end{aligned} \tag{30}$$

(4) The Expected Busy Cycle. Let the expected length of the idle period be denoted by $E(i=0)$. Applying the memoryless property of the Poisson process, the length of the idle period is the sum of N exponential random variables, each having a mean of $1/\lambda$. Hence, the expected length of the idle period is presented by $E(i=0) = N/\lambda$.

The busy cycle for the fully observable case is denoted by $E(C_N)$, which is the length of time from the beginning of completing all the customers in system to the beginning of the next completing of all the customers in system, which included vacation period, setup period, and busy period. From the relationship

$$\frac{E(i=0)}{E(C_N)} = P(i=0) = Np_{\text{ob}}(0,0), \tag{31}$$

we can get the expected length of the busy cycle $E(C_N) = 1/[\lambda p_{\text{ob}}(0,0)]$.

(5) The Social Benefit per Time. The social benefit per time unit as soon as all customers follow the equilibrium policy $(N, n_e(0), n_e(1))$ equals

$$\begin{aligned} S_{\text{ob}} &= \lambda R\left[1 - p_{\text{ob}}\left(1, n_e(1)+1\right) - p_{\text{ob}}\left(2, n_e(2)+1\right)\right] \\ &\quad - C\left[\sum_{n=0}^{N-1} np_{\text{ob}}(0,n) + \sum_{n=0}^{n_e(1)+1} np_{\text{ob}}(1,n)\right. \\ &\quad \left. + \sum_{n=1}^{n_e(2)+1} np_{\text{ob}}(2,n)\right]. \end{aligned} \tag{32}$$

(6) The Expected Cost Function. We next define the following:

C_h = holding cost per unit time for every customer present in the system;

C_s = cost incurred per busy cycle;

C_0 = cost per unit time for keeping the server off;

C_1 = setup cost per unit time for turning the server on;

C_2 = cost per unit time for keeping the server on and in operation;

C_3 = a fixed cost for every lost customer.

By using the definitions of all costs listed above, the expected cost function per unit time per customer is given by

$$\begin{aligned} &F_{\text{ob}}(N) \\ &= C_h L_N + \frac{C_s}{E(C_N)} + C_0 P(i=0) + C_1 P(i=1) \\ &\quad + C_2 P(i=2) \\ &\quad + \lambda C_3\left[p_{\text{ob}}\left(1, n_e(1)+1\right) + p_{\text{ob}}\left(2, n_e(2)+1\right)\right]. \end{aligned} \tag{33}$$

4. Equilibrium Strategies and Some Performance Measures for the Fully Unobservable System

We turn our attention to the fully unobservable case, where the customers do not observe the state and system size at all when they join the system. We will study the equilibrium arrival rate and stationary probabilities for the fully unobservable case as well as some interesting performance measures of this system.

4.1. Equilibrium Mixed Strategies and Some Stationary Probabilities for the Fully Unobservable Case. We will prove that there exists a mixed equilibrium strategy. An arriving customer joins the fully unobservable system with a certain probability q; then the effective arrival rate is $\lambda_0 = \lambda q$. If all customers adopt the same mixed strategy, then the system follows a Markov chain which is similar to that described in Figure 1. The state space S_{un} for the fully unobservable case is identical to the original state space S. In addition to substituting λ_0 for λ, the state transfer diagram is exactly the same as Figure 1.

Let $(p_{\text{un}}(i,n) : (i,n) \in S_{\text{un}})$ be the stationary distribution of the corresponding system. Let $p_{\text{un}}(0,0)$ be the probability that the system is empty.

Theorem 5. *In the fully unobservable $M/M/1$ queue with N policy, setup times, and $\sigma_0 \neq \rho_0$, in which the customers enter the system with probability q, the stationary probabilities of the system are given by*

$$p_{un}(0,n) = p_{un}(0,0), \quad n = 1, 2, \ldots, N-1, \tag{34}$$

$$p_{un}(1,n) = \sigma_0^{n-N+1} p_{un}(0,0), \quad n = N, N+1, \ldots, \tag{35}$$

$$p_{un}(2,n) = \frac{\rho_0}{1-\rho_0}\left(1-\rho_0^n\right) p_{un}(0,0), \quad n = 1, 2, \ldots, N, \tag{36}$$

$$p_{un}(2,n)=\frac{\rho_0\left\{(1-\rho_0)\sigma_0^{n+1-N}-\rho_0^{n+1-N}\left[1-\sigma_0+\rho_0^{N-1}(\sigma_0-\rho_0)\right]\right\}p_{un}(0,0)}{(1-\rho_0)(\sigma_0-\rho_0)}\quad n=N+1,N+2,\ldots, \tag{37}$$

$$p_{un}(0,0)=\frac{(1-\rho_0)(1-\sigma_0)}{N-(N-1)\sigma_0}, \tag{38}$$

where $\lambda_0=\lambda q$, $\rho_0=\lambda_0/\mu$, and $\sigma_0=\lambda_0/(\lambda_0+\theta)$.

Proof. The balance equations are presented below:

$$\lambda_0 p(0,0)=\mu p(2,1), \tag{39}$$

$$\lambda_0 p(0,n)=\lambda_0 p(0,n-1),\quad n=1,2,\ldots,N-1, \tag{40}$$

$$(\lambda_0+\theta)p(1,N)=\lambda_0 p(0,N-1), \tag{41}$$

$$(\lambda_0+\theta)p(1,n)=\lambda_0 p(1,n-1),\quad n=N+1,N+2,\ldots, \tag{42}$$

$$(\lambda_0+\mu)p(2,1)=\mu p(2,2), \tag{43}$$

$$(\lambda_0+\mu)p(2,n)=\lambda_0 p(2,n-1)+\mu p(2,n+1),\quad n=2,3,\ldots,N-1, \tag{44}$$

$$(\lambda_0+\mu)p(2,n)=\theta p(1,n)+\lambda_0 p(2,n-1)+\mu p(2,n+1),\quad n=N,N+1,\ldots, \tag{45}$$

$$\sum_{n=0}^{N-1}p(0,n)+\sum_{n=N}^{\infty}p(1,n)+\sum_{n=1}^{\infty}p(2,n)=1. \tag{46}$$

From (40), we easily get (34) in Theorem 5. From (41) and (42), we can get (35) in Theorem 5. From (39) and (43), we get $p_{\mathrm{un}}(2,1)$ and $p_{\mathrm{un}}(2,2)$. By substituting them into (44), we immediately get expression (36) in Theorem 5. Next, we put $p_{\mathrm{un}}(2,N-1)$ and $p_{\mathrm{un}}(2,N)$ into expression (45); we can get (37) in Theorem 5. By putting (34)–(37) into (46), we obtain the expression of $p_{\mathrm{un}}(0,0)$ at last. □

Theorem 6. *In the fully unobservable $M/M/1$ queue with N policy and setup times, where the customers enter the system with probability q, when a customer enters the system in state i ($i=0,1,2$), his expected sojourn time W_i is given below:*

$$\begin{aligned}W_0&=\frac{1}{\theta}+\frac{1}{2}\left(\frac{N+1}{\mu}+\frac{N-1}{\lambda_0}\right),\\ W_1&=\frac{N+1}{\mu}+\frac{1}{\theta}\left(1+\frac{\lambda_0}{\mu}\right),\\ W_2&=\frac{N}{2\mu}+\frac{1}{\mu-\lambda_0}+\frac{1}{\mu(1-\sigma_0)}-\frac{N}{2\mu\left[N-(N-1)\sigma_0\right]},\end{aligned} \tag{47}$$

and the average sojourn time of an arriving customer is

$$W(\lambda_0)=\frac{1}{\mu-\lambda_0}+\frac{1}{\theta}+\frac{(N-1)N\theta}{2\lambda(\theta N+\lambda_0)}. \tag{48}$$

Proof. Let $p_{\mathrm{un}}(i)$ be the probability that a customer arrives during state i, where $i=0,1,2$. From Theorem 5, we can easily get the steady state probabilities of the server:

$$\begin{aligned}P_{\mathrm{un}}(i=0)&=NP_{\mathrm{un}}(0,0),\\ P_{\mathrm{un}}(i=1)&=\sum_{n=N}^{\infty}p_{\mathrm{un}}(0,n)=\frac{\sigma_0 P_{\mathrm{un}}(0,0)}{1-\sigma_0},\\ P_{\mathrm{un}}(i=2)&=\sum_{n=1}^{N}p_{\mathrm{un}}(1,n)+\sum_{n=N+1}^{\infty}p_{\mathrm{un}}(1,n)=\rho_0.\end{aligned} \tag{49}$$

Let $p(n\mid i)$ be the stationary probability that the queue length is n conditional on observing the servers status which is i, where $i=0,1,2$ and $n=0,1,\ldots$. The conditional probabilities can be written by

$$\begin{aligned}p(n\mid 0)&=\frac{p_{\mathrm{un}}(0,n)}{p_{\mathrm{un}}(i=0)}=\frac{1}{N},\quad n=1,2,\ldots,N-1,\\ p(n+N\mid 1)&=\frac{p_{\mathrm{un}}(1,n+N)}{p_{\mathrm{un}}(i=1)}=(1-\sigma_0)\sigma_0^n,\quad n=0,1,\ldots,\\ p(n\mid 2)&=\frac{p_{\mathrm{un}}(2,n)}{p_{\mathrm{un}}(i=2)}=\frac{(1-\sigma_0)(1-\rho_0^n)}{N-(N-1)\sigma_0},\quad n=1,2,\ldots,N,\\ p(n+N\mid 2)&=\frac{p_{\mathrm{un}}(2,n+N)}{p_{\mathrm{un}}(i=2)}\\ &=\frac{(1-\sigma_0)\left\{(1-\rho_0)\sigma_0^{n+1}-\rho_0^{n+1}\left[1-\sigma_0+\rho_0^{N-1}(\sigma_0-\rho_0)\right]\right\}}{(\sigma_0-\rho_0)\left[N-(N-1)\sigma_0\right]},\quad n=1,2,\ldots.\end{aligned} \tag{50}$$

Note that if the queue length is n and the server is busy, the expected sojourn time of the new arriving customer is denoted as $w(n\mid 2)$, which is the expected service time of the $n+1$ customer in the system, that is, $w(n\mid 2)=(n+1)/\mu$, which means the expected sojourn time upon seeing a busy server is

$$w(n\mid 2)=\frac{n+1}{\mu},\quad n=1,2,\ldots. \tag{51}$$

The expected sojourn time upon seeing a setup server is

$$w(n\mid 1)=\frac{1}{\theta}+\frac{n+N+1}{\mu},\quad n=0,1,\ldots. \tag{52}$$

The expected sojourn time upon seeing the server on N policy is

$$w(n \mid 0) = \frac{N-(n+1)}{\lambda_0} + \frac{1}{\theta} + \frac{n+1}{\mu}, \quad n = 0, 1, \ldots, N-1. \tag{53}$$

Then the expected sojourn time W_i $(i = 0, 1, 2)$ is given as follows:

$$\begin{aligned} W_0 &= \sum_{n=0}^{N-1} w(n \mid 0)\, p(n \mid 0) \\ &= \frac{1}{\theta} + \frac{1}{2}\left(\frac{N+1}{\mu} + \frac{N-1}{\lambda_0}\right), \\ W_1 &= \sum_{n=0}^{\infty} w(n+N \mid 1)\, p(n+N \mid 1) \\ &= \frac{N+1}{\mu} + \frac{1}{\theta}\left(1 + \frac{\lambda_0}{\mu}\right), \\ W_2 &= \sum_{n=1}^{\infty} w(n \mid 2)\, p(n \mid 2) \\ &= \sum_{n=1}^{N} \frac{n+1}{\mu} \frac{(1-\sigma_0)(1-\rho_0^n)}{N-(N-1)\sigma_0} \\ &\quad + \sum_{n=1}^{\infty} \frac{n+N+1}{\mu} p(n+N \mid 2) \\ &= \frac{N}{2\mu} + \frac{1}{\mu-\lambda_0} + \frac{1}{\mu(1-\sigma_0)} \\ &\quad - \frac{N}{2\mu[N-(N-1)\sigma_0]}. \end{aligned} \tag{54}$$

Then the average sojourn time of an arriving customer is

$$\begin{aligned} W(\lambda_0) &= W_0 P(i=0) + W_1 P(i=1) + W_2 P(i=2) \\ &= \frac{1}{\mu-\lambda_0} + \frac{1}{\theta} + \frac{(N-1)N\theta}{2\lambda_0(\theta N + \lambda_0)}. \end{aligned} \tag{55}$$

□

Next, according to the value of N, we divide N into two cases as follows.

Case 1. When $N = 1$, the system becomes the fully unobservable $M/M/1$ queue with setup times and $\lambda < \mu$. The average sojourn time reduces to

$$W(\lambda_0) = \frac{1}{\theta} + \frac{1}{\mu-\lambda_0}. \tag{56}$$

We consider a tagged customer at his arrival instant, if he decides to enter his expected net benefit which is

$$R - C\left(\frac{1}{\theta} + \frac{1}{\mu-\lambda_0}\right), \quad \text{where } \lambda_0 = \lambda q. \tag{57}$$

When $R \in (C/\mu + C/\theta, C/(\mu-\lambda) + C/\theta)$, we find that $R - C(1/\theta + 1/(\mu-\lambda q)) = 0$ has a unique root in $(0,1)$. When $R > C/(\mu-\lambda) + C/\theta$, we have $R - C(1/\theta + 1/(\mu-\lambda q)) > 0$ for any $q \in [0,1]$. The best response for an arriving customer is to join the system. Then we have the following proposition.

Proposition 7. *In the fully unobservable $M/M/1$ queue with setup times and $\lambda < \mu$, there exists a unique mixed equilibrium strategy "enter with probability q_e," where the vector q_e is given by*

$$q_e = \begin{cases} \frac{1}{\lambda}\left(\mu - \frac{C}{R - C/\theta}\right), & \text{if } R \in \left(\frac{C}{\mu} + \frac{C}{\theta}, \frac{C}{\mu-\lambda} + \frac{C}{\theta}\right); \\ 1, & \text{if } R \in \left(\frac{C}{\mu-\lambda} + \frac{C}{\theta}, +\infty\right). \end{cases} \tag{58}$$

This conclusion is consistent with Theorem 5 in [13].

Case 2. When $N \geq 2$, we have

$$W''(\lambda_0) = (N-1)\left[\frac{1}{\lambda_0^3} - \frac{1}{(\theta N + \lambda_0)^3} + \frac{2}{(\mu-\lambda_0)^3}\right] > 0, \quad \lambda_0 \in (0, \mu), \tag{59}$$

which means that $W(\lambda_0)$ is strictly convex in λ_0 and $\lim_{\lambda_0 \to 0+} W(\lambda_0) = \lim_{\lambda_0 \to \mu-} W(\lambda_0) = +\infty$, and then there is a unique minimum point $\widetilde{\lambda} \in (0, \mu)$, so that $W(\widetilde{\lambda})$ is the minimum sojourn time and $\widetilde{\lambda}$ satisfies the following equation:

$$W'(\lambda_0) = \frac{N-1}{2(\theta N + \lambda_0)^2} - \frac{N-1}{2\lambda_0^2} + \frac{1}{(\mu-\lambda_0)^2} = 0. \tag{60}$$

It is clear that when $R < CW(\widetilde{\lambda})$, there is no positive equilibrium arrival rate; when $R = CW(\widetilde{\lambda})$, there exists one positive equilibrium arrival rate $\widetilde{\lambda} \in (0, \mu)$ iff $\widetilde{\lambda} < \lambda$; when $R > CW(\widetilde{\lambda})$, that is, $R \in (CW(\widetilde{\lambda}), +\infty)$, the equation

$$R = C\left[\frac{1}{\mu-\lambda_0} + \frac{1}{\theta} + \frac{(N-1)N\theta}{2\lambda_0(\theta N + \lambda_0)}\right] \tag{61}$$

has two real roots $\lambda_1 \in (0, \widetilde{\lambda})$ and $\lambda_2 \in (\widetilde{\lambda}, \mu)$. We analyze the strictly convex $W(\lambda_0)$ carefully and then find that the equilibrium with λ_1 is unstable, for, with any small increase of the arrival rate, the expected waiting time decreases and more customers will arrive in the system; this will further increase λ_1. The equilibrium with λ_2 and λ is stable; that is, if there is a small perturbation to them, the system will converge back to them. We denote the stable equilibrium arrival rate as λ^*; then we have the following proposition.

Proposition 8. *(a) If $R < CW(\widetilde{\lambda})$, there is no positive equilibrium arrival rate. (b) If $R = CW(\widetilde{\lambda})$, there is one positive*

Table 1: Numerical examples for $\mu = 0.42$ and $\lambda_2 < \lambda$.

$u = 0.42, R = 30, \theta = 0.2, C = 1, \lambda = 0.4$										
N	1	2	3	4	5	6	7	8	9	10
λ_1	$q_e = 0.95$	0.0211146	0.0418168	0.0626556	0.0837833	0.105319	0.127408	0.150264	0.174238	0.2
λ_2		0.378885	0.377214	0.375112	0.372582	0.369567	0.365949	0.361532	0.355973	0.34861
λ^*	0.38	0.378885	0.377214	0.375112	0.372582	0.369567	0.365949	0.361532	0.355973	0.34861

Table 2: Numerical examples for $\mu = 0.52$ and $\lambda_1 < \lambda < \lambda_2$.

$u = 0.52, R = 30, \theta = 0.2, C = 1, \lambda = 0.4$										
N	1	2	3	4	5	6	7	8	9	10
λ_1	$q_e = 1$	0.0206732	0.0408608	0.0610673	0.0814019	0.101923	0.122683	0.14374	0.165172	0.187081
λ_2		0.479226	0.478046	0.476574	0.474841	0.472846	0.470565	0.467952	0.464943	0.461439
λ^*	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4

Table 3: Numerical examples for $\mu = 1$ and $\lambda_1 < \lambda < \lambda_2$.

$u = 1, R = 30, \theta = 0.2, C = 1, \lambda = 0.4$										
N	1	2	3	4	5	6	7	8	9	10
λ_1	$q_e = 1$	0.0391792	0.0583989	0.0583989	0.0776041	0.096822	0.116065	0.135342	0.154658	0.174019
λ_2		0.95934	0.958828	0.958828	0.958219	0.957533	0.95678	0.955963	0.955084	0.954143
λ^*	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4

equilibrium arrival rate $\widetilde{\lambda} \in (0, \mu)$ *iff* $\widetilde{\lambda} < \lambda$. *(c) If* $R > CW(\widetilde{\lambda})$, *there exists positive equilibrium arrival rate*

$$\lambda_e = \begin{cases} \lambda_1 \text{ or } \lambda_2, & \text{if } \lambda_2 < \lambda; \\ \lambda_1 \text{ or } \lambda, & \text{if } \lambda_1 < \lambda < \lambda_2; \\ \text{nonexistent}, & \text{if } \lambda < \lambda_1, \end{cases} \tag{62}$$

and the stable positive equilibrium arrival rate

$$\lambda^* = \begin{cases} \lambda_2, & \text{if } \lambda_2 < \lambda; \\ \lambda, & \text{if } \lambda_1 < \lambda < \lambda_2; \\ \text{nonexistent}, & \text{if } \lambda < \lambda_1, \end{cases} \tag{63}$$

where $\widetilde{\lambda}$ *is the positive solution of* λ_0 *in (60) and* $\lambda_1 \in (0, \widetilde{\lambda})$ *and* $\lambda_2 \in (\widetilde{\lambda}, \mu)$ *are the positive solutions of* λ_0 *in (61).*

According to Proposition 7 and case (c) of Proposition 8, we give concrete numerical forms as shown in Tables 1, 2, and 3.

4.2. The System Performance Measures for the Fully Unobservable Case. Our objective is to give some interesting performance measures of the fully unobservable system and determine the optimum value of the control threshold N by numerical computations.

(1) The Expected Number of Customers. Let L_{un} denote the expected number of customers in the (N, q) equilibrium strategy. Then we obtain

$$L_{\text{un}} = \frac{N(N-1)\, p_{\text{un}}(0,0)}{2} + \sum_{n=N}^{+\infty} n p_{\text{un}}(1,n) + \sum_{n=1}^{+\infty} n p_{\text{un}}(2,n). \tag{64}$$

(2) The Probability of the Server in Idle, Setup, and Busy Steady State as Follows. Consider

$$P(i=0) = N p_{\text{un}}(0,0); \quad P(i=1) = \sum_{n=N}^{+\infty} p_{\text{un}}(1,n); \quad P(i=2) = \sum_{n=1}^{+\infty} p_{\text{un}}(2,n). \tag{65}$$

(3) The Expected Busy Cycle. Similar to the fully observable case, the expected length of the idle period is given by $E(i = 0) = N/\lambda_0$ and the expected length of the busy cycle $E(C_N) = 1/[\lambda_0 p_{\text{un}}(0,0)]$.

(4) The Social Benefit per Time. We assume the stable equilibrium arrival rate $\lambda_0 = \lambda^*$ and all customers follow

the equilibrium strategy (N, q); then the social benefit per time unit when all customers follow the equilibrium equals

$$\begin{aligned} S_{\text{un}} &= \lambda_0 \left[P(i=0)B(0) + P(i=1)B(1) + P(i=2)\right. \\ &\quad \left.\cdot B(2)\right] = \lambda_0 N p_{\text{un}}(0,0)\left\{R - C\left[\frac{1}{\theta}\right.\right. \\ &\quad \left.\left. + \frac{1}{2}\left(\frac{N+1}{\mu} + \frac{N-1}{\lambda_0}\right)\right]\right\} + \lambda_0 \frac{\sigma_0 p_{\text{un}}(0,0)}{(1-\sigma_0)}\left\{R\right. \\ &\quad \left. - C\left[\frac{N+1}{\mu} + \frac{1}{\theta}\left(1 + \frac{\lambda_0}{\mu}\right)\right]\right\} + \lambda_0 \rho_0 \left\{R\right. \\ &\quad \left. - C\left(\frac{N}{2\mu} + \frac{1}{\mu - \lambda_0} + \frac{1}{\mu(1-\sigma_0)}\right.\right. \\ &\quad \left.\left. - \frac{N}{2\mu\left[N - (N-1)\sigma_0\right]}\right)\right\}. \end{aligned} \tag{66}$$

(5) The Expected Cost Function. By using the definition of each cost listed before, the expected cost function per unit time per customer is given by

$$\begin{aligned} F_{un}(N) &= C_h L_N + \frac{C_s}{E(C_N)} + C_0 P(i=0) \\ &\quad + C_1 P(i=1) + C_2 P(i=2) + (\lambda - \lambda_0) C_3. \end{aligned} \tag{67}$$

5. Numerical Examples

In this section, we obtain some numerical experiments to show the different effects of the fully observable and unobservable information systems. We let $R = 30, C = 1, \lambda = 0.4$, and $\theta = 0.2$ in all the figures. For the fully observable case, from inequality (3), we have $N \le \lfloor 1 + \lambda R/C - \lambda/\theta - \lambda/\mu \rfloor = 10$ for any $\mu > 0.4$. We vary the values of positive integer N from 1 to 10 in Figures 3, 4, and 5.

In Figure 3, we select $\mu = 0.42, 0.52, 1$ and vary the values of positive integer N from 1 to 10. We find that, for the fully observable case, the mean customers L_{ob} are an increasing function of N and a decreasing function of service rate μ. For the fully unobservable case, when $\mu = 0.42$, from Table 1, we find that the stable arrival rate λ^* is reducing by increasing N, which means more customers are balking the system, so the system size is reducing by increasing N; when $\mu = 0.52$ and 1, we find that the stable arrival rate λ^* is a constant value and does not change by varying the values of N from 1 to 10; the system size which is an increasing function of N on account of more customers is blocked. When μ and N are fixed, we find, as for $\mu = 0.42$ and 0.52, the system size of fully observable L_{ob} is smaller than corresponding fully unobservable L_{un}. When $\mu = 1$, L_{ob} and L_{un} are almost equal. It is clear that when customers know the system size and server state, they may be more rational to decide whether to enter the system, which brings more effective regulation to the system size. When the service rate is very larger than arrival rate and the service reward is generous, which means that customers get fast service, good reward and very few customers balk the system, then such good advantages inevitably lead to the disclosures of information (the server state and system size) having a little effect on the system size.

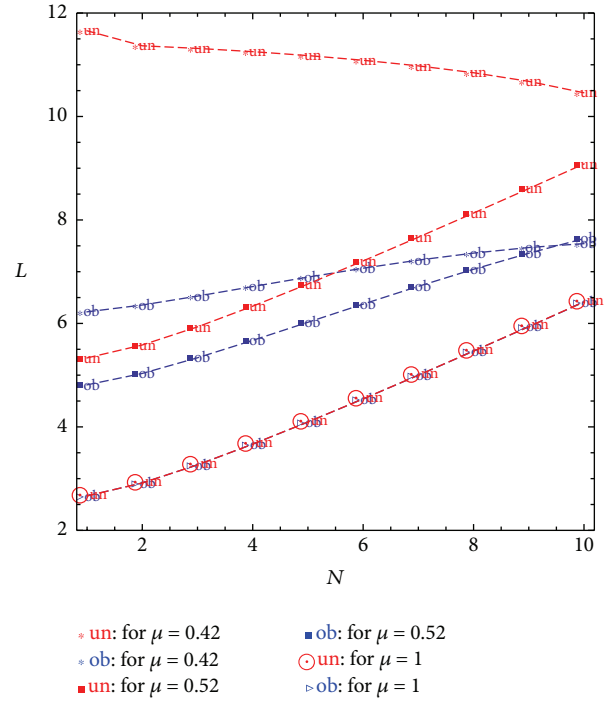


FIGURE 3: System size L versus N for different μ.

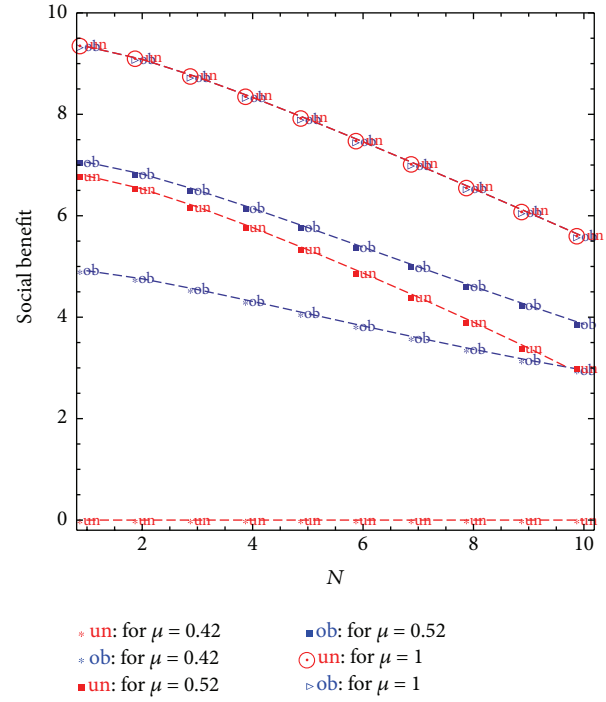


FIGURE 4: Social benefit S versus N for different μ.

In Figure 4, we find that, along with the increase of N, the social benefits S_{ob} and S_{un} are on the decline. S_{ob} is bigger than S_{un} when $\mu = 0.42, 0.52$ and select the same N; S_{ob} is almost equal to S_{un} when $\mu = 1$ and select the same N.

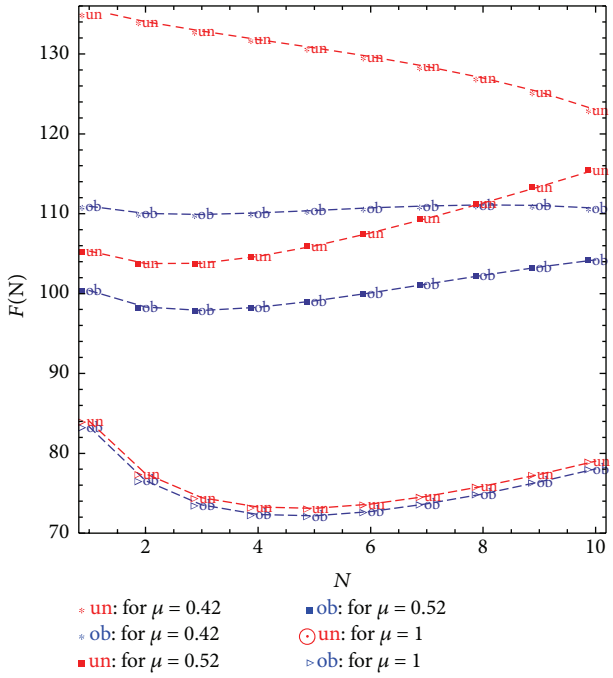


FIGURE 5: Cost function $F(N)$ versus N for different μ.

For unobservable case, when $\mu = 0.42$, from Table 1, we know that the equilibrium arrival strategy taken by all the customers is the cost of sojourn time which equals the service reward, which leads to $S_{un} = 0$.

In Figure 5, we still select $\mu = 0.42, 0.52, 1$ and vary the values of positive integer N from 1 to 10. We further fix the following cost parameters: $C_h = 5$, $C_s = 200$, $C_0 = 10$, $C_1 = 50$, $C_2 = 80$, and $C_3 = 30$. We find that when $\mu = 0.42, 0.52$ and selecting the same N, the value of $F_{ob}(N)$ is always bigger than $F_{un}(N)$; when $\mu = 1$ and selecting the same N, $F_{ob}(N)$ is almost equal to $F_{un(N)}$. If we select appropriate N, we can reduce the expected cost for all the fully observable and unobservable cases.

6. Conclusions

In this paper, we studied the equilibrium behavior of customers in $M/M/1$ queue with N policy and server setup times. We provide fully observable and fully unobservable scenarios with respect to system size and system state provided to arriving customers and derived the equilibrium balking strategies for each case. We mainly study how the N policy affects system size, social benefit, and system cost under the condition of equilibrium strategies taken by fully observable and fully unobservable arriving customers.

Conflict of Interests

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

Acknowledgments

This research is partially supported by CPSF (2015M572327), NSFC (71131003, 71371075, and 71271089), and NSFA (KJ2013B162, KJ2013A194, and KJ2014A174). It is also supported by Chaohu University Scientific Research Fund. The authors would like to sincerely thank the anonymous referees and editors for their valuable comments and suggestions which are very helpful for them to improve the presentation of the paper.

References

[1] P. Naor, "The regulation of queue size by levying tolls," *Econometrica*, vol. 37, no. 1, pp. 15–24, 1969.

[2] A. Economou and S. Kanta, "Equilibrium customer strategies and social-profit maximization in the single-server constant retrial queue," *Naval Research Logistics*, vol. 58, no. 2, pp. 107–122, 2011.

[3] R. Hassin and M. Haviv, *To Queue or Not to Queue: Equilibrium Behavior in Queueing Systems*, Kluwer Academic, Boston, Mass, USA, 2003.

[4] H. Chen and M. Frank, "Monopoly pricing when customers queue," *IIE Transactions*, vol. 36, no. 6, pp. 569–581, 2004.

[5] M. Yadin and P. Naor, "Queueing systems with a removable service station," *Journal of the Operational Research Society*, vol. 14, no. 4, pp. 393–405, 1963.

[6] K.-H. Wang and J.-C. Ke, "A recursive method to the optimal control of an M/G/1 queueing system with finite capacity and infinite capacity," *Applied Mathematical Modelling*, vol. 24, no. 12, pp. 899–914, 2000.

[7] K.-H. Wang, T.-Y. Wang, and W. L. Pearn, "Optimal control of the N policy $M/G/1$ queueing system with server breakdowns and general startup times," *Applied Mathematical Modelling*, vol. 31, no. 10, pp. 2199–2212, 2007.

[8] P. Guo and R. Hassin, "Strategic behavior and social optimization in Markovian vacation queues," *Operations Research*, vol. 59, no. 4, pp. 986–997, 2011.

[9] P. Guo and Q. Li, "Strategic behavior and social optimization in partially-observable Markovian vacation queues," *Operations Research Letters*, vol. 41, no. 3, pp. 277–284, 2013.

[10] G. Choudhury, "An $M^X/G/1$ queueing system with a setup period and a vacation period," *Queueing Systems*, vol. 36, pp. 23–38, 2000.

[11] W. Bischof, "Analysis of $M/G/1$-queues with setup times and vacations under six different service disciplines," *Queueing Systems*, vol. 39, no. 4, pp. 265–301, 2001.

[12] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, "A survey of scheduling problems with setup times or costs," *European Journal of Operational Research*, vol. 187, no. 3, pp. 985–1032, 2008.

[13] A. Burnetas and A. Economou, "Equilibrium customer strategies in a single server Markovian queue with setup times," *Queueing Systems*, vol. 56, no. 3-4, pp. 213–228, 2007.

[14] A. Economou and S. Kanta, "Equilibrium balking strategies in the observable single-server queue with breakdowns and repairs," *Operations Research Letters*, vol. 36, no. 6, pp. 696–699, 2008.

[15] P. Chen and Y. Zhou, "Equilibrium balking strategies in the single server queue with setup times and breakdowns," *Operational Research*, vol. 15, no. 2, pp. 213–231, 2015.

[16] F.-C. Jiang, D.-C. Huang, C.-T. Yang, and F.-Y. Leu, "Lifetime elongation for wireless sensor network using queue-based approaches," *The Journal of Supercomputing*, vol. 59, no. 3, pp. 1312–1335, 2012.

[17] D. H. Lee and W. S. Yang, "The N-policy of a discrete time *Geo*/*G*/*1* queue with disasters and its application to wireless sensor networks," *Applied Mathematical Modelling*, vol. 37, no. 23, pp. 9722–9731, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 276460, 9 pages
http://dx.doi.org/10.1155/2015/276460

Research Article

Minimum Time Approach to Emergency Collision Avoidance by Vehicle Handling Inverse Dynamics

Wang Wei, Bei Shaoyi, Yang Hui, Wang Yongzhi, and Zhang Lanchun

School of Automotive and Traffic Engineering, Jiangsu University of Technology, Changzhou, Jiangsu 213001, China

Correspondence should be addressed to Wang Wei; nuaawangwei@126.com

Received 8 August 2014; Revised 29 October 2014; Accepted 16 November 2014

Academic Editor: Yi-Chung Hu

Vehicle driving safety is the urgent key problem to be solved of automobile independent development while encountering emergency collision avoidance with high speed. And it is also the premise and one of the necessary conditions of vehicle active safety. A new technique of vehicle handling inverse dynamics which can evaluate the emergency collision avoidance performance is proposed. Based on optimal control theory, the steering angle input and the traction/brake force imposed by driver are the control variables; the minimum time required to complete the fitting biker line change is the control object. By using the improved direct multiple shooting method, the optimal control problem is converted into a nonlinear programming problem that is then solved by means of the sequential quadratic programming. The simulation results show that the proposed method can solve the vehicle minimum time maneuver problem, and can compare the maneuverability of two different vehicles that complete fitting biker line change with the minimum time and the correctness of the model is verified through real vehicle test.

1. Introduction

With the continuous development of the automobile industry, the number of car accidents grows accordingly, especially in traffic accidents involving pedestrians and cyclists. In some cases, car accidents can be seen as a collision between vehicles and obstacles. And then the emergency avoidance problem is proposed to avoid the accidents. Today, people pay more and more attention to the problem of high-speed emergency avoidance [1, 2]. When vehicles traveling at high speed meet obstacles, the driver will often choose the emergency brake parking or the bypass passing to avoid obstacles. Both the maneuvers require the driver to avoid obstacles in minimum time [3].

The research methods of vehicle handling dynamics usually include open-loop and closed-loop method. The two methods are called "forward problem" method of vehicle handling dynamics research. Open-loop research method does not consider the function of the driver's feedback and obtains vehicle response under the condition of mathematical model of vehicle and driver input. But closed-loop method obtains vehicle motion which follows the ideal path based on driver vehicle closed-loop control system model [4]. In order to avoid building driver model which is a difficult and important problem for vehicle model, the method of vehicle handling inverse dynamics is proposed [5–7]. The vehicle handling inverse dynamics can be reversed to obtain the driver's handling input based on the known model and vehicle motion (vehicle response). Then, the vehicle handling inverse dynamics can analyze what kind of handing is easily accepted by the driver with the safest and most rapid way [8–10]. Google is developing self-driving technology that combines data collected by sensors installed on a car with existing mapping software to speed up, brake, and steer to a destination. First, the driver's handling input was obtained in the Google car. According to the handling input, self-drive mode controls the car. So Google car apply the principle of the vehicle handling inverse dynamics. The Mercedes Benz system of PRE-SAFE and BAS-PLUS are designed to help to avoid accidents. The effectiveness is a measure for the efficiency, with which a safety system succeeds in achieving this target within its range of operation in vehicle. The process is also the application of the vehicle handling inverse dynamics [11].

In emergency avoidance research, the vast majority of research focused on the shortest path in the process of emergency avoidance. Sundar and Shiller (1997) proposed a method producing the shortest path based on the Hamilton-Jacobi-Bellman equation in a cluttered environment. The method attributed the emergency avoidance problem of shortest distance to the optimal control problem of shortest time, generated the shortest path through the function of negative gradient, and achieved good results [12]. Hattori et al. (2006) optimized the vehicle trajectory control for obstacle avoidance problem. A new control algorithm for obstacle avoidance within the shortest possible distance is proposed [2]. Mukai et al. [13] transformed the problem of generating an optimal path without a collision between an automobile and obstacles as a mixed integer programming problem [2]. However, the minimum time required to complete the emergency avoidance was very little to research. The maneuvers under the emergency avoidance require the driver to avoid obstacles in minimum time. So the minimum time approach to emergency collision avoidance is very important. In the paper, the inverse dynamics method was introduced to research the minimum time problem of the vehicle emergency collision avoidance.

Vehicle handling inverse dynamics can evaluate the driver's handling input by the specified handling performance and improve the performance of high-speed vehicle emergency avoidance. The handling performance of different vehicle can be compared with the most efficient way by the vehicle handling inverse dynamics [14–16].

In the paper, the optimal control theory is used in the field of vehicle handling inverse dynamics. In order to simplify the problem, the ideal driver handling inputs are considered without consideration of driver response lag and the forward-looking role.

2. Vehicle Steering Motion Model

2.1. The Mathematical Model of Vehicle Steering Wheel Torque Input. Assuming tire cornering properties in the linear range and considering rotational inertia of the steering system, the vehicle steering motion model is simplified as shown in Figure 1, which is a linear vehicle model with four degrees of freedom (DOF). The four degrees are lateral movement, horizontal pendulum movement, longitudinal motion, and steering system turning. A 4 DOF vehicle steering motion model is built as shown in Figure 1. The differential equations of motion are expressed as

$$\begin{aligned}
\dot{v} &= -u\omega_r + \frac{F_{yf}\cos\delta + F_{yr} + F_{xf}\sin\delta}{m},\\
\dot{\omega}_r &= \frac{aF_{yf}\cos\delta - bF_{yr} + aF_{xf}\sin\delta}{I_z},\\
\dot{u} &= v\omega_r + \frac{F_{xf}\cos\delta - F_{yf}\sin\delta + F_{xr} - F_f - F_w}{m},\\
\dot{\delta} &= p\\
\dot{p} &= -\frac{k_1\xi_1}{I_w u}v - \frac{k_1\xi_1 a}{I_w u}\omega_r + \frac{(k_1\xi_1 - k_w)}{I_w}\delta - \frac{c_w}{I_w}p + \frac{T_{sw}i}{I_w},
\end{aligned} \tag{1}$$

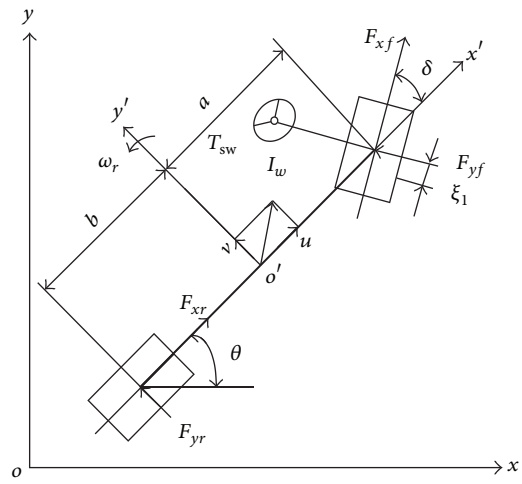


Figure 1: 4 DOF vehicle steering model.

where v is the lateral velocity, u is the longitudinal velocity, ω_r is the yaw rate, m is the total mass of vehicle, I_z is the vehicle moment of inertia around the vertical axis, I_w is the steering system moment of inertia, ξ_1 is the returnable arm of front wheel, p is the state variables, c_w is the resistance coefficient of steering system, k_w is the composite stiffness of steering system, i is transmission ratio of steering system, T_{sw} is the torque of steering wheel, a, b are the distance from the whole vehicle centroid to front and rear axle, k_1, k_2 are the comprehensive cornering stiffness of the former and rear wheels, δ is the rotation angle of the former and rear wheels, F_{yf} is the cornering force of the front wheel, F_{yr} is the cornering force of the rear wheel, F_{yr} is the driving/braking force of the front wheel (when $F_{xf} \geq 0$, F_{xf} is driving force; when $F_{xf} < 0$, F_{xf} is braking force), F_{xr} is the rear wheel driving/braking force, F_f is the rolling resistance ($F_f = mgf$; f is the rolling resistance coefficient), and F_w is the air resistance ($F_w = C_D A(3.6u)^2/21.15$; C_D is the air resistance coefficient and A is the windward area). The vehicle runs on smooth surfaces, no slope resistance. In order to simplify the problem, the air resistance and rolling resistance are considered, not considering acceleration resistance.

If driving force/braking force is considered to impact the cornering force, it is

$$\begin{aligned}
F_{yf} &= k_1\left(\frac{v + a\omega_r}{u} - \delta\right)\sqrt{1 - \left(\frac{F_{xf}}{F_{zf}}\right)^2 + \left(\frac{F_{xf}}{k_1}\right)^2}\\
F_{yr} &= k_2\left(\frac{v - b\omega_r}{u}\right)\sqrt{1 - \left(\frac{F_{xr}}{F_{zr}}\right)^2 + \left(\frac{F_{xr}}{k_2}\right)^2},
\end{aligned} \tag{2}$$

where φ is the friction coefficient of pavement, F_{zf} is the vertical force of front wheel, and F_{zr} is the vertical force of rear

wheel. Take the longitudinal load transfer into consideration; it is

$$F_{zf} = \frac{mgb - \left(F_{xf} + F_{xr}\right) h_g}{a+b},$$
$$F_{zr} = \frac{mga + \left(F_{xf} + F_{xr}\right) h_g}{a+b}, \tag{3}$$

where h_g is the height of vehicle mass center.

The x, y coordinates of the vehicle mass center in the xoy coordinate system have the following relations:

$$\dot{x} = u\cos\theta - v\sin\theta$$
$$\dot{y} = v\cos\theta + u\sin\theta. \tag{4}$$

2.2. The Optimal Control Model of Steering Wheel Torque Input. Control variable $Z(t)$ is the steering wheel torque $T_{\mathrm{sw}}(t)$ and wheel driving force/braking force $F_{xf}(t)$; the control goal is the minimum time through a given path. Therefore, the performance indicators of time are

$$J(Z) = \int_{t_0}^{t_e} \mathrm{d}t, \tag{5}$$

where t_0, t_e are the initial time and terminal time.

According to (1), state equation can be expressed as

$$\dot{\mathbf{X}} = f\left[X(t), Z(t)\right]. \tag{6}$$

In the equation, the state variables are

$$X(t) = \left[v(t)\ \ \omega_r(t)\ \ u(t)\ \ \delta(t)\ \ p(t)\ \ x(t)\ \ y(t)\ \ \theta(t)\right]^{\mathrm{T}}. \tag{7}$$

Longitudinal velocity u is bounded by the vehicle maximum speed, lateral displacement is bounded by track around the border, the angle of the steering wheel δ_{sw} is bounded by driver's physiological limit, and control variable F_{xf} is bounded by road adhesion. When the vehicle is under a front wheel driving, it is

$$F_{xf} \le \frac{\varphi mgb}{a + b + \varphi h_g},$$
$$F_{xr} = 0. \tag{8}$$

When the vehicle is under a braking force and the front and rear wheels are assumed in lock state, it is

$$F_{xf} \ge -\frac{\varphi mg\left(b + \varphi h_g\right)}{a+b},$$
$$F_{xr} = \frac{a - \varphi h_g}{b + \varphi h_g} F_{xf}. \tag{9}$$

Control variable F_{xf} is bounded by the maximum driving force which is provided by the power transmission system.

According to the connection between engine speed and the velocity of vehicles and the connection between the engine output torque and the driving force, the regulation between the maximum driving force and the velocity of vehicles can be obtained by the engine external characteristic curve.

According to the literature [17], the constraints preventing rollover in the course of the vehicle driving are

$$\frac{u^2\delta}{(a+b)\left(1 + Ku^2\right) g} \le \frac{L}{2h_g}, \tag{10}$$

where L is wheel tread and K is stability factor.

All the constraints are shown by the following equation:

$$\psi\left[\mathbf{X}(t), Z(t)\right] = 0. \tag{11}$$

2.3. The Transformation of State Variables. In the process of the vehicle tracking the desired trajectory, the ultimate elapsed time is difficult to determine. In order to solve this problem conveniently, the free terminal time can be transformed into the fixed terminal time for optimal control problem with the following ways.

Longitudinal displacement variable $\overline{x}$ which is defined unitization is

$$\overline{x} = \frac{x - x_0}{x_e - x_0} \quad (0 \le \overline{x} \le 1), \tag{12}$$

where x_0 is the initial longitudinal displacement and x_e is the terminal longitudinal displacement.

According to (4) and the time derivative of (12), the following equation can be obtained by taking a derivative with respect to time t in (12). Consider

$$\frac{\mathrm{d}\overline{x}}{\mathrm{d}t} = \frac{1}{x_e - x_0}\dot{x} = \frac{u\cos\theta - v\sin\theta}{x_e - x_0} \quad (0 \le \overline{x} \le 1). \tag{13}$$

Make state variable $\overline{\mathbf{X}}(\overline{x}) = \left[v(\overline{x})\ \ \omega_r(\overline{x})\ \ u(\overline{x})\ \ \delta(\overline{x})\ \ p(\overline{x})\ \ y(\overline{x})\ \ \theta(\overline{x})\right]^{\mathrm{T}}$.

According to (13), performance index of (5) can be transformed into

$$\mathrm{d}t = \frac{x_e - x_0}{u\cos\theta - v\sin\theta}\mathrm{d}\overline{x} \quad (0 \le \overline{x} \le 1)$$
$$J(Z) = \int_{\overline{x}_0}^{\overline{x}_e} \lambda\, \mathrm{d}\overline{x}, \tag{14}$$

where $\lambda = (x_e - x_0)/(u\cos\theta - v\sin\theta)$.

Similarly, (6) is

$$\frac{\mathrm{d}\overline{\mathbf{X}}}{\mathrm{d}\overline{x}} = \lambda f\left[\overline{\mathbf{X}}(\overline{x}), Z(\overline{x})\right]. \tag{15}$$

2.4. Nonlinear Programming Method of Improved Direct Multiple Shooting. The state variable, control variable, and time of nodes are assumed at the same time in the direct multiple shooting algorithm. It will increase variable numbers of the transformed nonlinear programming problem, thus making it more difficult to get the answer. Therefore, this paper puts

forward an improved direct multiple shooting method; in other words, only control variables of nodes are assumed.

(1) The Original Problem Is Converted into a Fixed Terminal Time Mayer Problem [18]. The state space expands into $n+1$ dimensions; new state variables are introduced to satisfy the following two equations:

$$\dot{\overline{X}}_{n+1} = \lambda, \tag{16}$$

$$\overline{X}_{n+1}(0) = 0. \tag{17}$$

The performance index of (14) can be translated into

$$J(Z) = \lambda \overline{X}_{n+1}(1). \tag{18}$$

Therefore, as long as (16) is incorporated in to the system state equation (15) and (17) is merged into constraint equation (11), the original problem is translated into standard Mayer problem with given terminal.

(2) The Optimal Control Problem Is Transformed into Finite Dimensional Nonlinear Programming Problem. The optimization method used in the paper is one of the sequential quadratic programming method (SQP): Wilson-Han-Powell method. The method is based on the common Lagrange-Newton method. The basic theory of SQP converts the nonlinear programming problem to a series of quadratic programming problems, so it is called SQP. Specifically, the approximate solution x_k and approximate multiplier vector λ_k were assumed to be known when the kth iteration starts. In this way, the kth quadratic programming subproblem P_k can be given, the new approximate solution x_{k+1} can be obtained by solving the problem P_k, and the corresponding Lagrange multiplier vector λ_{k+1} is determined too. The above-mentioned process is repeated until the approximate optimal solution of nonlinear programming problem is obtained. Assuming $d_k = x_{k+1} - x_k$, then, getting x_{k+1} by solving P_k can be converted to getting the d_k by solving the subproblem P_k.

Considering the common nonlinear constrained optimal control problem,

$$\begin{aligned} \min_{x \in R^n} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0, \quad i \in E \\ & c_i(x) \geq 0, \quad i \in I, \end{aligned} \tag{19}$$

where $f(x)$, $c_i(x)$ are all real-valued continuous functions and at least one of them is nonlinear, $E = \{1, 2, \ldots, m_e\}$, $E = \{1, 2, \ldots, m_e\}$. The subproblem is constructed. Consider

$$\begin{aligned} \min_{d \in R^n} \quad & g_k^T d + \frac{1}{2} d^T B_k d \\ \text{s.t.} \quad & a_i(x_k)^T d + c_i(x_k) = 0, \quad i \in E \\ & a_i(x_k)^T d + c_i(x_k) \geq 0, \quad i \in I. \end{aligned} \tag{20}$$

In the above equations, $A(x_k) = [a_1(x_k), \ldots, a_m(x_k)] = \nabla c(x_k)^T$, g_k is the gradient of $f(x)$ in the x_k point, and B_k is the approximation of Hesse matrix of Lagrange function. The solution of above subproblem is d_k; the d_k is used as the search direction of kth iteration in the method of Wilson-Han-Powell. It is the descent direction of penalty function.

The procedures of sequential quadratic programming are given as follows.

(1) Give

$$x_1 \in R^n, \quad \sigma > 0, \quad \delta > 0, \quad B_1 \in R^{n \times n}, \quad \varepsilon \geq 0, \quad k = 1. \tag{21}$$

(2) One gets d_k by solving the above subproblem. If $\|d_k\| \leq \varepsilon$, then stop the iteration. Solve $\alpha_k \in [0, \delta]$, which makes

$$P(x_k + \alpha_k d_k, \sigma) \leq \min_{0 \leq \alpha \leq \delta} P(x_k + \alpha_k d_k, \sigma) + \varepsilon_k. \tag{22}$$

(3) $x_{k+1} = x_k + \alpha_k d_k$; calculating B_{k+1}, $k = k + 1$; return to step (2).

In formula (22), the penalty function $P(x, \sigma)$ is the precise penalty function of L_1; ε_k is a nonnegative series and satisfies the following condition:

$$\sum_{k=1}^{\infty} \varepsilon_k < +\infty. \tag{23}$$

B_{k+1} is obtained by using quasi-Newton formula: take

$$\begin{aligned} s_k &= x_{k+1} - x_k, \\ y_k &= \nabla f(x_{k+1}) - \nabla f(x_k) - \sum_{i=1}^{m} (\lambda_k)_i \left[\nabla c_i(x_{k+1}) - \nabla c_i(x_k)\right]. \end{aligned} \tag{24}$$

Then calculate B_{k+1} by using BFGS correction formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}. \tag{25}$$

For the optimal control problem of time-varying system in this paper, it can be converted to the finite dimensional nonlinear programming problem by using the improved direct multiple shooting method.

(1) The interval $\overline{x} \in [0, 1]$ is divided into n uniform. $n+1$ nodes are obtained.

(2) A set of vectors e_j $(j = 0, 1, \ldots, n-1)$ are introduced as estimated values of control variable at the node. Control variable values between nodes are gotten by linear interpolation of two adjacent values. If the node place control variables e_j are known, each state variable can be gotten one by one by integration. Thus $\overline{X}_{n+1}(1)$ can be obtained, and then performance index is gotten. Therefore, it can be argued that the solution of differential equation and performance indicators are only the function of each node control variable.

The gating finite dimensional nonlinear programming problem can be solved by using the sequential quadratic programming (using the fmincon function in the optimization toolbox of MATLAB).

Table 1: Two vehicles parameter list.

Parameter	Model A	Model B
m/kg	1500	1265
I_z/(kg·m^2)	2080	1800
a/m	1.185	1.170
b/m	1.283	1.195
k_1/(N·rad^{-1})	−60533	−60042
k_2/(N·rad^{-1})	−110185	−109295
i	20	20
I_w/(kg·m^2)	17.53	16.38
c_w/(N·m·s·rad^{-1})	160	140
k_w/(N·m·rad^{-1})	0	0
ξ_1/m	0.028	0.021
φ	0.8	0.8
h_g/m	0.53	0.53
i_g	0.97	0.914
i_0	4.12	4.38
r/m	0.289	0.280
u/km·h^{-1}	72	72

i_g is the transmission gear ratio, i_0 is the main reducer gear ratio, and r is the wheel radius.

3. Numerical Simulation

The biker line performance of two vehicles is researched. The vehicle specific parameter values are shown in Table 1. The steering wheel torque is controlled between ±8 N·m. The initial time is

$$\begin{aligned}&\left[v(0)\ \ \omega_r(0)\ \ u(0)\ \ \delta(0)\ \ p(0)\ \ x(0)\ \ y(0)\ \ \theta(0)\right]^{\mathrm{T}}\\&\quad=\left[0\ \ 0\ \ 20\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\right]^{\mathrm{T}}.\end{aligned}\tag{26}$$

The size of biker line test road is shown in Figure 2. Parameter value in Figure 2 is $s_0 = L = 2u$, $s = 3u$. Benchmarking width is $B = 2.46$ m.

In the actual driving process, driver's ideal target track should be as shown in Figure 3. It is a low order continuous smooth curve. Three-order curve of a continuous first derivative is gotten after three spline fits. Consider

$$f(t) = \begin{cases} 0 & t \in t_1 \\ \dfrac{B(1.0+\sin\omega t)}{2.0} & t \in t_2 \\ B\cos\omega t & t \in t_3 \\ \dfrac{B(-1.0+\sin\omega t)}{2.0} & t \in t_4 \\ 0 & t \in t_5. \end{cases}\tag{27}$$

Parameter Model A and Model B are as in Table 1.

As is shown in Table 1, the curb weight of A vehicle was bigger than B vehicle. A vehicle was better than B vehicle in the configuration and space.

After 14 iterations, the minimum time in which model A passes the biker line after optimization is 15.7 s. After 16 iterations, the minimum time in which model B passes the biker line after optimization is 16.2 s. Therefore, the minimum time in which model A passes the biker line after optimization is shorter than that of model B. Figures 4~7 show simulation results of some state variables and control variables in the process of biker line when $u = 72$ km/h.

Figure 2: Biker line test road.

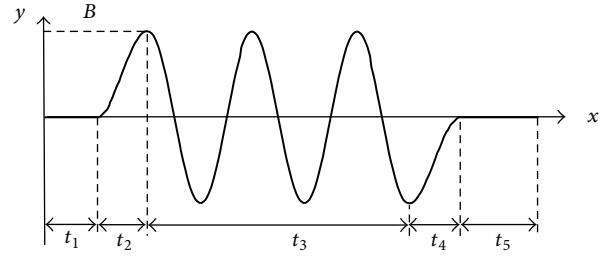


Figure 3: Fitting biker line.

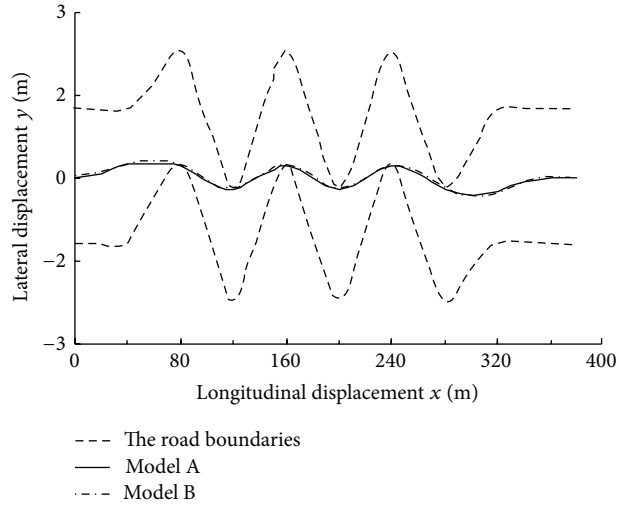


Figure 4: Lateral displacement simulation results.

Figure 4 shows the simulation results about lateral displacements of two types of models. It can be seen that the two kinds of models' lateral displacement are almost coincidence. The motion law for vehicle between the road boundaries is that vehicles move almost in straight line.

Figure 5 shows the simulation results about steering wheel torque of two vehicles. It can be seen from the several steering wheel torque amplitudes that the steering wheel torque amplitude of vehicle A is larger than that of vehicle B.

Figure 6 shows the simulation results about the wheel driving force of two vehicles. It can be seen that the driving force decreases at first and then increases in the serpentine

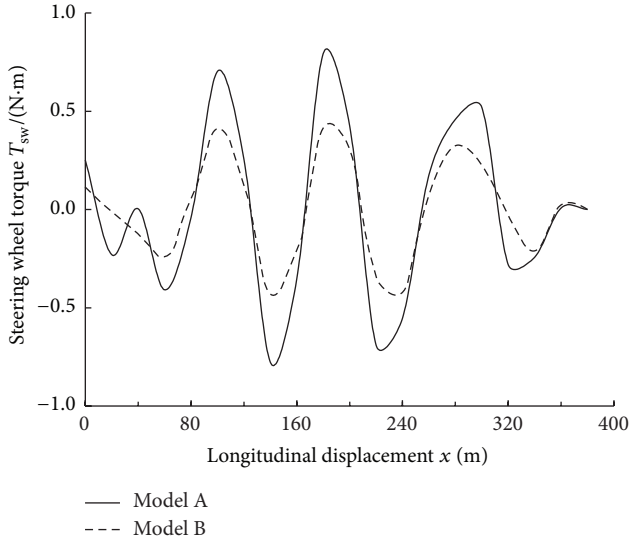


Figure 5: The steering wheel torque simulation results.

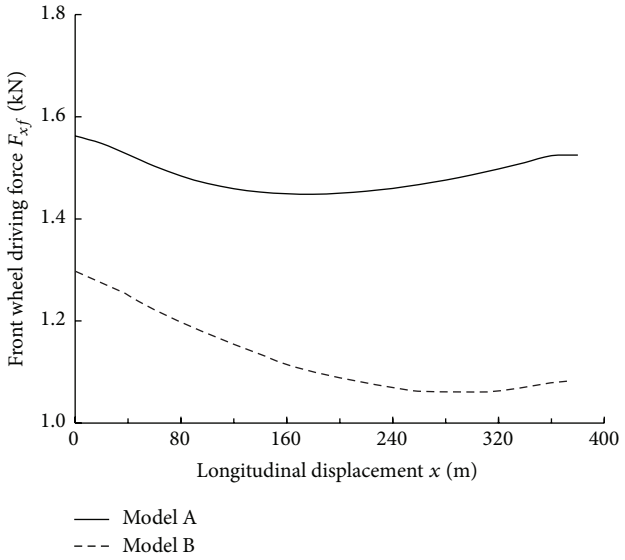


Figure 6: Wheel driving force simulation results.

line performance process; it also can be seen that the vehicle A's driving force is larger than that of the vehicle B.

Figure 7 shows the two models' longitudinal velocity simulation results. It can be seen that in serpentine line performance process, Model A increases rapidly from 20 m/s to 30.7 m/s and Model B increases rapidly from 25 m/s to 29.2 m/s. Therefore, the acceleration performance of Model A is better than that of B.

When the vehicle travels at 25 m/s high speed initially, after 14 iterations, the minimum time in which Model A passes the biker line after optimization is 14.9 s. After 18 iterations, the minimum time in which model A passes the biker line after optimization is 15.2 s. Therefore, when the vehicle run at the fast speed, it drive through the serpentine in short time. But as the vehicle's speed increases, the driver's

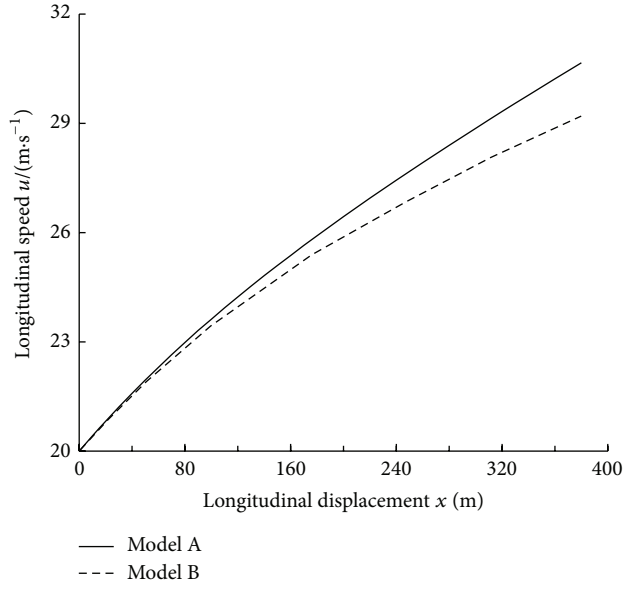


Figure 7: Longitudinal velocity simulation results.

burden increases and the safety reduces. The driver has to reduce vehicle's speed to a certain level to ensure his safety before the vehicle passes the biker line.

4. Experimental Verification

In this paper, two types of off-road vehicles mentioned above are used to test vehicle handling stability. Real vehicle test is very dangerous in high speed. In order to consider the driver's safety, the method of pavement design point is taken in the test.

The test procedures are as follows.

(1) In the test site, stake position marker is designed as in Figure 5.

(2) Connect the test instruments; switch instruments power on in order towarm the instruments to normal operating temperature.

(3) The vehicle passes the test section with an initial speed of 72 km/h. Running over the marker is not allowed in the running process. At the same time, the time history curve of the measured variables (steering wheel angle and longitudinal velocity) is recorded by the computer.

(4) Repeat steps (3) process 12 times (the times of press the marker is not considered). Two vehicle types' experimental data are obtained by the same test methods above if the vehicle type is changed.

The test site is built as shown in Figure 8.

The experimental procedures and protocols are built as shown in Figure 9.

12 groups of test time were, respectively, 17.8 s, 18.1 s, 17.9 s, 18.5 s, 18.8 s, 18.3 s, 17.9 s, 18.3 s, 18.2 s, 18.9 s, 19.0 s, and 18.8 s. Due to considering a lot of factors, such as driver's reaction

(a) The bollard of biker line

(b) Car running track

(c) The instrument of steering wheel torque

(d) DEWESoft data signal acquisition system

Figure 8: Vehicle biker line test.

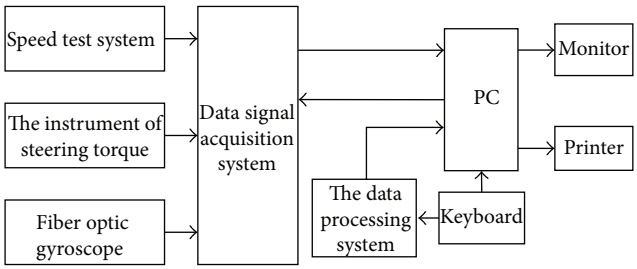


Figure 9: Experimental procedures and protocols.

time and road conditions, the experimental test time was generally longer than the time of optimal control. The mean and standard deviation were, respectively, 18.27 s and 0.3743.

Comparison between the simulation value and the experimental value is shown in Figures 10 and 11. As shown in Figures 10 and 11, there are some errors between simulation value and experimental value, mainly because every driver's subjective feelings and driving skills is different. In addition, the test instrument also has some errors. But the change tendency of simulation value and experimental values is consistent. So the correctness of the optimal control model is proved.

5. Conclusion

In the field of automotive engineering, many researchers are focusing on the development of self-driving technologies. Self-driving vehicles promise to bring a number of benefits to society, including prevention of road accidents, optimal fuel usage, comfort, and convenience. Vehicle handling inverse dynamics is form of the self-driving technologies. The steering wheel torque can be obtained by the vehicle handling inverse dynamics and used to determine the vehicle steering problems in the emergency collision avoidance. So the vehicle handling inverse dynamics can promote the self-driving vehicle development.

In this paper, minimum time approach to emergency collision avoidance is researched by the method of vehicle handling inverse dynamics. Firstly, the optimal control model of the vehicle emergency collision avoidance problem was established. And then the optimal control problem was changed into a nonlinear programming problem using the improved direct multiple shooting method. Finally, the transformed nonlinear programming problem was solved by using sequential quadratic programming method. The correctness of the optimal control model is verified by using real vehicle test. The results show that this method can successfully solve the minimum time problem of vehicle emergency

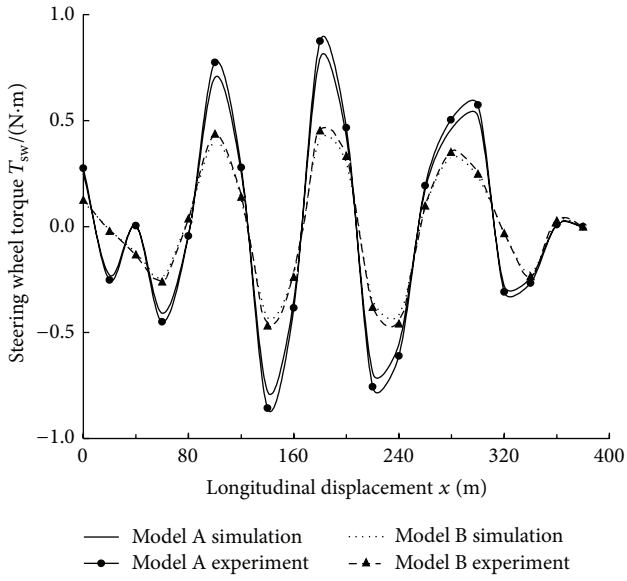


Figure 10: The steering wheel torque simulation results.

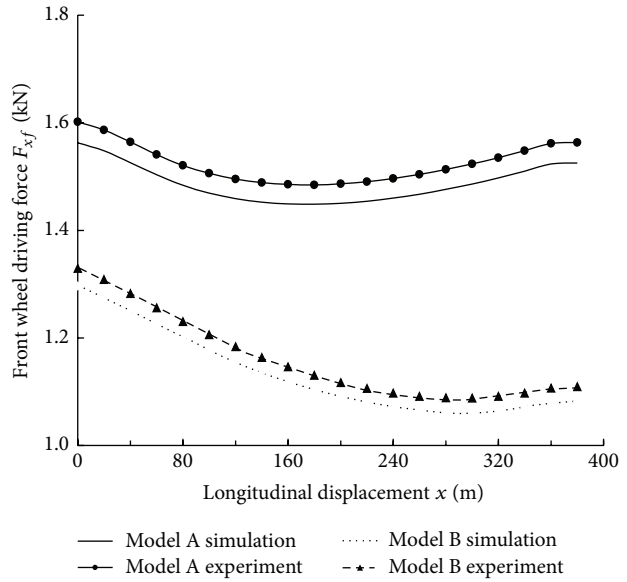


Figure 11: Wheel driving force simulation results.

collision avoidance and compare different vehicles in the minimum time through a given path control performance. It can provide guidance for the self-drive research. Intelligent vehicle driving also has certain reference value.

Conflict of Interests

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

Acknowledgments

This work was supported in part by the National Science Foundation of China (Grant no. 51305175) and the National Science Foundation of JiangSu Province (Grant no. BK2012586).

References

[1] A. Mas, F. Merienne, and A. Kemeny, "Lateral control assistance and driver behavior in emergency situations," *Advances in Transportation Studies*, pp. 149–158, 2011.

[2] Y. Hattori, E. Ono, and S. Hosoe, "Optimum vehicle trajectory control for obstacle avoidance problem," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 5, pp. 507–512, 2006.

[3] J.-M. Park, D.-W. Kim, Y.-S. Yoon, H. J. Kim, and K.-S. Yi, "Obstacle avoidance of autonomous vehicles based on model predictive control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 223, no. 12, pp. 1499–1516, 2009.

[4] H. Hatwal and E. C. Mikulcik, "Some inverse solutions to an automobile path-tracking problem with input control of steering and brakes," *Vehicle System Dynamics*, vol. 15, no. 2, pp. 61–71, 1986.

[5] J. Andreasson and T. Bunte, "Global chassis control based on inverse vehicle dynamics models," *Vehicle System Dynamics*, vol. 44, no. 1, pp. 321–328, 2006.

[6] C.-S. Liu and H. Peng, "Inverse-dynamics based state and disturbance observers for linear time-invariant systems," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 124, no. 3, pp. 375–381, 2002.

[7] J. Cho, J. C. Principe, D. Erdogmus, and M. A. Motter, "Modeling and inverse controller design for an unmanned aerial vehicle based on the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 445–460, 2006.

[8] C.-F. Lin and Y. Chuang, "Energy management strategy and control laws of an inverse differential gear hybrid vehicle," *World Electric Vehicle Journal*, vol. 4, no. 1, pp. 98–103, 2011.

[9] I. Han and S.-U. Park, "Inverse analysis of pre- and post-impact dynamics for vehicle accident reconstruction," *Vehicle System Dynamics*, vol. 36, no. 6, pp. 413–433, 2001.

[10] F. Boyer and S. Ali, "Recursive inverse dynamics of mobile multibody systems with joints and wheels," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 215–228, 2011.

[11] H. Schittenhelm, "Design of effective collision mitigation systems and prediction of their statistical efficiency to avoid or mitigate real world accidents," in *Proceedings of the 32nd FISITA World Automotive Congress*, vol. 7, pp. 14–23, September 2008.

[12] S. Sundar and Z. Shiller, "Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 305–310, 1997.

[13] M. Mukai, T. Kawabe, H. Nishira, Y. Takagi, and Y. Deguchi, "On vehicle path generation method for collision avoidance using mixed integer programming," in *Proceedings of the 16th IEEE International Conference on Control Applications*, pp. 1371–1375, Singapore, October 2007.

[14] J. D. Trom, M. J. Vanderploeg, and J. E. Bernard, "Application of inverse models to vehicle optimization problems," *Vehicle System Dynamics*, vol. 19, no. 2, pp. 97–110, 1990.

[15] J. P. M. Hendrikx, T. J. J. Meijlink, and R. F. C. Kriens, "Application of optimal control theory to inverse simulation of car

handling," *Vehicle System Dynamics*, vol. 26, no. 6, pp. 449–461, 1996.

[16] J. Sridhar and H. Hatwal, "Comparative study of four wheel steering models using the inverse solution," *Vehicle System Dynamics*, vol. 21, no. 1, pp. 1–18, 1992.

[17] J. Bernard and M. Pickelmann, "An inverse linear model of a vehicle," *Vehicle System Dynamics*, vol. 15, no. 4, pp. 179–186, 1986.

[18] D. Kraft, "Algorithm 733: TOMP—fortran modules for optimal control calculations," *ACM Transactions on Mathematical Software*, vol. 20, no. 3, pp. 262–281, 1994.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 738578, 13 pages
http://dx.doi.org/10.1155/2015/738578

Research Article

Configuration, Deployment, and Scheduling Models for Management and Optimization of Patrol Services

Bin Yang, Zhi-Hua Hu, and Jing-Xian Zhou

Logistics Research Center, Shanghai Maritime University, Shanghai 200135, China

Correspondence should be addressed to Zhi-Hua Hu; zhhu@shmtu.edu.cn

Received 30 September 2014; Revised 15 November 2014; Accepted 27 November 2014

Academic Editor: Kishin Sadarangani

This paper presents a decision support system (DSS) and its models for patrol service center (PSC). PSC plays an important role in public security and emergency management. The configuration, deployment, and scheduling of resources of PSC are important for improving the efficiency of patrol-related resources, service quantity, and emergency response capability. A series of decision-making models of the DSS are studied. First, the criteria and models are proposed for configuring and deploying PSCs; second, three types of models for incremental, direct, and redeployment optimization are built in views for decisions aiming at PSC configuration, deployment, and scheduling problems; third, considering three typical patrol-related service scenarios (alarm assignment, main road blockade, and besiege program), three scheduling models are built, respectively, for PSC-related service and coordination of multiple PSCs. This work contributes to the literature on patrol services and network optimization problems in the following aspects: based on a series of models, a DSS framework is designed for PSCs; the models are formulated for resource management and scheduling upon geography information system; coordination strategies among close PSCs are incorporated into decision models. These features are examined in integration manners. The assessment criteria and optimization models studied in the paper are beneficial for building DSSs for PSC.

1. Introduction

Police patrolling is regarded as one of the best well-known practices for implementing public-safety preventive policies towards the combat of an assortment of urban crimes [1]. In China, patrol police provides emergency service for local public security and contingent events. A patrol department is a police unit created primarily for the purpose of overseeing and enforcing traffic safety compliance on roads and highways. There are many managerial problems for the configuration, deployment, and scheduling of patrol-related resources responding to contingent events. To implement these functions more effectively, some patrol service centers (PSCs) are gradually established in major traffic arteries and densely inhabited districts. Due to the shortage of police resources, how can the PSCs be deployed, how can the area of each PSC be assigned, and how can the patrol-related resource be scheduled are practical problems faced by patrol management departments.

Activated by the requirements on PSC information management systems in China and the problems studied in the literature, a decision support system (DSS) for PSC with the following decision-making problems is identified: the configuration, deployment, reconfiguration, and redeployment of PSCs; coordinative response for serious contingent events; emergency blockade for serious suspects; and regional besiege program for serious suspects. The above problems, their optimization models, and decision strategies should be studied based on spatial information. In this study, a general DSS framework is proposed for PSC.

In the literature, some models have been proposed for patrol-elated optimization problems. Coupe and Blake [2], D'Amico et al. [3], Lau et al. [4], and Lou et al. [5] deal with the configuration and deployment problems of patrol-related resources in strategic levels. Keskin et al. [6] and Yin [7, 8] studied the scheduling problems of patrol-related resources and tasks. Various models were studied independently such that the practical values of them are limited. The following

literature studied the optimization problems of patrol-related resources in strategic levels. Pal and Sinha [9] evaluated freeway service patrol system configurations by using simulation. Bodily [10] proposed a decision-making model for a resource allocation problem in the design of service areas for police mobile units. Using multiattribute utility theory, alternative designs were evaluated according to the preferences for efficiency and equality of service of three interest groups: citizens, police, and administrators. Meaningful measures of inequality were developed and an algorithm was created for generating improved sector designs. Carroll and Laurin [11] defined police patrol zones by computer simulation. The criterion is to maximize the amount of time available for patrol duties; the constraints are the current average time to respond to a call and a limit on the number of patrol cars. Kern [12] allowed customized sector design and deployment strategies to be tested by simulation. The simulation model reported allows the administrator to study the performance of several complex dispatching tactics including multiple unit dispatching, preemption, and intersector dispatching. D'Amico et al. [3] proposed a simulated annealing algorithm for districting the police commands. Coupe and Blake [2] studied the effects of patrol workloads of and response strength to burglary emergencies. Curtin et al. [13] presented a new method for determining efficient spatial distributions of police patrol areas. This method employs a traditional maximal covering formulation and an innovative backup covering formulation to provide alternative optimal solutions to police decision makers. Assunção and Furtado [14] describe a heuristic method based on graph partitioning algorithms with the purpose of improving the demarcation of areas for police patrolling. Lou et al. [5] investigated the problem of deploying freeway service patrols to detect, respond to, and clear traffic incidents in deterministic and stochastic settings. Lau et al. [4] developed a mathematical model to improve the deployment of security guards in a company located in south China for the purpose of immediate response to complaints.

The second type of topics is about scheduling patrol-related resources and tasks. Khattak et al. [15] developed an approach to help determine the most beneficial locations for patrol deployment by using expanded placement criteria. Results of the research were incorporated into a decision support tool that allows easy planning and operational assessment of candidate sites by comparing performance values between sites, modeling the effect of patrol services, and estimating their key potential benefits. Yin [7] proposed a min-max bilevel programming model to find an optimal fleet allocation solution for freeway service patrols. The model minimizes the maximal traveling time when incidents may incur, which was solved by a heuristic algorithm. Reis et al. [1] devised an evolutionary multiagent-based simulation tool to assist police managers in the design of effective police patrol route strategies, where the high crime-density regions are well covered by routine patrol surveillance. Keskin et al. [6] addressed the problem of determining the routes of state troopers to maximize the coverage of highway spots with high frequencies of crashes. A specific mixed-integer linear programming (MIP) model was developed for this problem under time restrictions and budget limitations. An algorithm incorporating heuristics based on local search and tabu search is designed to solve the model. Yin [8] formulated a mixed-integer nonlinear programming model based on scenario analysis for the fleet allocation problem to achieve faster response and reduced clearance time in patrol-related management background. Chaiken and Dormont [16, 17] developed advanced patrol car allocation models, which were updated to include multiple dispatch queuing [18], and the validity was tested with generally positive results given the limitations of a model dependent on human behavior [19]. Further, Sacks [20] examined the spatial deployment of police cars with respect to four criteria: response time, workload balance, patrol frequency, and interdistrict dispatches.

Comparing to the above literature, the contributions of this study include the following points. First, a DSS framework is designed for PSCs. Second, mathematical models are proposed for resource management and scheduling upon geography information system (GIS). Third, coordination strategies among close PSCs are incorporated into decision models.

The paper is organized as follows. In Section 2, a framework of DSS for PSC is proposed. In Section 3, the assessment criteria and a series of decision models in the DSS PSCs are studied. In Section 4, the models are demonstrated by using the sample data of a city in China. In Section 5, the conclusion and the suggestions of future work are discussed.

2. System Framework

The configuration, deployment, and scheduling of PSCs depend on three types of information: (1) positions of roads and residents; (2) real-time positions and status of patrol-related resources, for example, cars and phones; (3) the urban public security system and other management systems and the collected and transferred information by the 110 call center. The accuracy and intervals of the three types of real-time information impose a great effect on the efficiency and performance of scheduling the PSCs. Moreover, the historical information of these types is the premise for deploying PSCs. As shown in Figure 1, the DSS for PSC (PSC DSS) is built upon spatial data from GIS, the databases of the city and PSCs, and other information management systems. PSC DSS connects with the public security inspection and management system and 110 call center, to acquire the service demands which are sent to specific PSC by wireless communication.

The PSC DSS in Figure 1 includes two parts of decision models: scheduling models which usually focus on real-time operations and deployment models which are commonly periodically performed and have strong strategic significance. For scheduling models, the service area assignment is also performed periodically. According to the alarm demands assigned to specific PSC, specific scheduling model is chosen to be performed. Emergent blockade and besiege programs represent two scheduling levels. Emergent blockade aims for the main crossroads in a specific areas in the city when some serious suspects are escaping; the besiege program is to schedule the patrol-related resources in the urban area for the escaping serious suspects. These two scheduling

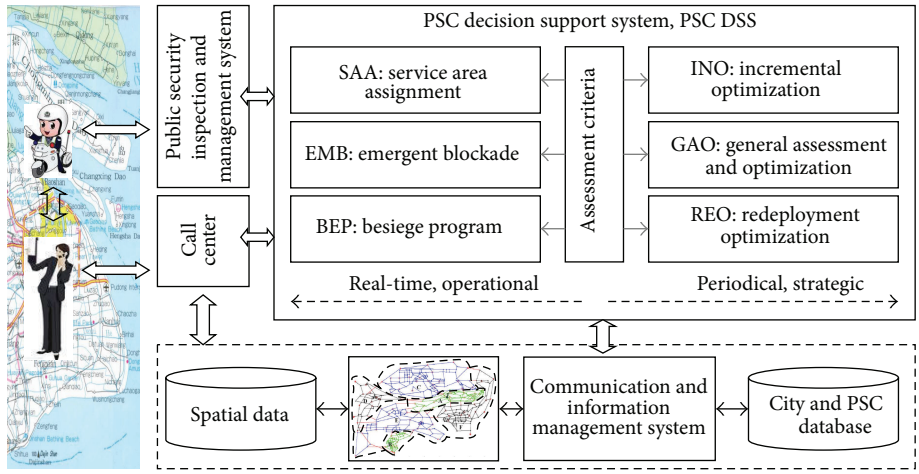


Figure 1: The proposed framework of PSC DSS.

models coordinate the PSCs. Three models are designed for deploying PSCs. First, according to the patrol service rules and general requirements of patrol-related resource configuration, a system of assessment criteria is proposed. Second, based on the current deployment of PSCs, new PSCs are incrementally built to enhance the service capability in the areas with more alarms. The third is about redeploying PSCs. It is apparent that both the incremental deployment and redeployment aim at finding optimal solutions under the assessment criteria.

3. Models

In the following, the problems proposed above and their decision models proposed in Figure 1 are studied.

3.1. Service Area Assignment. The model of service area assignment (SAA) is to allocate crossroads to PSCs after the alarm demands are all ascribed to crossroads by GIS. SAA has two typical scenarios. First, the crossroads with workloads represented by alarms are periodically assigned to PSCs. Second, the service demand triggered by an alarm is sent to a specific PSC by 110 call center according to the location of the alarm. The processes in the above scenarios should utilize the information that can be returned by the following systems: urban information management system, urban GIS, and 110 call center.

SAA assigns nodes in the traffic network to PSC. The fundamental data that can be utilized include the crossroad node set that can be obtained by topology analysis based on GIS and the cost matrix among these nodes; the set of PSCs and the nodes of all PSCs; the alarm frequency and the workload of each node that can be calculated from the patrol service records; finally the length of time promised for responding to alarm by patrol service after the alarm is accepted. The length is three minutes in many cities in China. According to the above information, the objective is to build a map from road nodes to PSCs, where a PSC can serve a set of nodes.

The related sets, parameters, and decision variables are defined as follows. The road node set is denoted by NOD. The PSC set is S, $S \subseteq$ NOD. $CT_{a,b}$ sets the traveling time between two nodes $a, b \in$ NOD. F_n sets the alarm frequency of node $n \in$ NOD. The response time promised by PSC is denoted by MT, and commonly $MT = 3$. Four decision variable groups are defined below. First, $x_{s,n} = 1$, if the alarm from the $n \in$ NOD is processed by the PSC located at $s \in$ NOD; or else zero. Second, stp_s defines the workload of the PSC ($s \in S$). Third, ntr_n denotes the response time of the alarm proposed at the node $n \in$ NOD. The fourth group of decision variables is defined for improving the readability of the model: *stpma* is the maximal workload of all PSCs; *stpmi* is the minimal workload of all PSCs; *ntrma* is the maximal weighted response time of all nodes; and *ntrall* is the total weighted response time of all nodes.

The objective functions reflect the optimization and balance of the benefits of PSC and residents. For a PSC, the total service workloads represented by the service times and service distances are minimized. Among different PSCs, the workloads should be balanced. For the residents, the response time to alarm is most important. The promise of three minutes for response should be tried to be satisfied. In the following, (1)–(3) are designed to cater for these objectives.

(1) Balance the total working time computed by weighted alarm frequencies.

This objective is to minimize the difference between the maximal workload and the minimal workload (1), where the maximal workload and the minimal workload are calculated by (8) and (9). From the views of optimal response and service quantity, two other objectives are defined in the following.

(2) Minimize the maximal weighted response time of nodes.

Equation (2) minimizes the longest weighted response time, where *ntrma* is defined further by (7).

(3) Minimize the total weighted response time.

Equation (2) is defined in the view of minimizing the service workload of a single node. From another view, (3) minimizes the weighted response service workloads of all nodes, where *ntrall* is defined by (10).

Before defining the variables used in the objective functions, the response time for node's alarm and the total weighted service time for each PSC are defined. In (4), according to the assignment of nodes to PSCs, the response time for each node is defined. According to the promise of response time within three minutes, if it is a hard constraint, the response time of each node should satisfy (5), where MT is the limit.

Based on the constraints ((4)–(6)), the objectives ((1)–(3)) are further defined by (7)–(10). The maximal response time of node is defined by (7). The maximal workload of PSC is computed by (8), whereas (9) calculates the minimal workloads for PSC. The total weighted response time of all nodes is further summed by (10).

Additionally, the constraint in (11) sets a restriction that the alarm from a node must and can only be assigned to a PSC. Moreover, for simplicity, the node of a PSC should be assigned to this PSC (12):

$$\min f_1 = stpma - stpmi, \tag{1}$$

$$\min f_2 = ntrma, \tag{2}$$

$$\min f_3 = ntrall, \tag{3}$$

$$ntr_n = \sum_{s\in S} \left(x_{s,n} \cdot CT_{s,n}\right), \quad \forall n \in \mathrm{NOD}, \tag{4}$$

$$ntr_n \le MT, \quad \forall n \in \mathrm{NOD}, \tag{5}$$

$$stp_s = \sum_{n\in \mathrm{NOD}} \left(x_{s,n} \cdot CT_{s,n} \cdot F_n\right), \quad \forall s \in S, \tag{6}$$

$$ntrma \ge ntr_n, \quad \forall n \in \mathrm{NOD}, \tag{7}$$

$$stpma \ge stp_s, \quad \forall s \in S, \tag{8}$$

$$stpmi \le stp_s, \quad \forall s \in S, \tag{9}$$

$$ntrall = \sum_{n\in \mathrm{NOD}} \left(ntr_n \cdot F_n\right), \tag{10}$$

$$\sum_{s\in S} x_{s,n} = 1, \quad \forall n \in \mathrm{NOD}, \tag{11}$$

$$x_{s,s} = 1, \quad \forall s \in S. \tag{12}$$

3.2. Emergent Blockade. Emergent blockade (EMB) is usually a solution to deal with or prevent the accidents in the following cases: when there is big accident that happened in the area, and the car flow and people flow will impede solving the problem, for example, a traffic accident by dangerous chemicals or virus; the flows of cars and people themselves will incur accidents, for example, big fog climate which requires blockading highways. The model EMB builds a map between the main road nodes and the patrol-related resources.

The quantity of main road nodes is presumed to be less than the quantity of PSCs. EMB assigns main road nodes to PSCs. Therefore, three set types are involved: the road nodes, PSCs, and the main road nodes. The parameters mainly include a matrix of traveling time between road nodes. The decision variables include the mapping relations between the main road nodes and PSCs. Besides NOD and S, E defines the node set attached to the PSCs, $E \subseteq$ NOD. The variable $x_{s,e} = 1$, if the main road node $e \in E$ is blockaded by PSC $s \in S$; or else 0. et_e is the time to blockade the main road node $e \in E$. *etma* represents the maximal time to blockade the main road nodes, whereas *etall* is the total time to blockade all main road nodes.

The objectives of blockade should emphasize urgency. The urgency here has two meaning levels: the blockade should be finished as soon as possible; the blockade cost should be minimized by optimizing the assignment of PSCs to main road nodes. *etma* has been defined to denote the maximal time of blockade, and *etall* is the total time of blockade. Then, two objectives are defined ((13) and (14)) to minimize *etma* and *etall*. They are further computed by (18) and (19).

The constraint functions in EMB include two parts: assigning PSCs to the main road nodes and the variables used in the objectives:

(1) every main road node should be assigned a PSC for blockade (15);

(2) every PSC cannot take charge of more than one main road node (16);

(3) the time to blockade a given main road node is computed by x(17);

(4) the maximal time of blockade is limited ((18) and (19));

$$\min f_1 = etma, \tag{13}$$

$$\min f_2 = etall, \tag{14}$$

$$\sum_{s\in S} x_{s,e} = 1, \quad \forall e \in E, \tag{15}$$

$$\sum_{e\in E} x_{s,e} \le 1, \quad \forall s \in S, \tag{16}$$

$$et_e = \sum_{s\in S} \left(x_{s,e} \cdot CT_{s,e}\right), \quad \forall e \in E, \tag{17}$$

$$etma \ge et_e, \quad \forall e \in E, \tag{18}$$

$$etall = \sum_{e\in E} et_e. \tag{19}$$

3.3. Besiege Program. Designing besiege program (BEP) is a dynamic optimization problem. Considering a serious case happens at node $P \in$ NOD, the suspect has escaped by car after the alarm was received after three minutes. When the escape direction of the suspect cannot be determined, if the PSC $s \in S$ decides to blockade the suspect at the node $a \in$ NOD, (20) should be satisfied to ensure that the patrol policemen can reach not before the suspect reaches it.

In another aspect, the time of finishing besiege can determine the coverage of the besiege program. For example, if it is wished to finish besiege within five minutes, then every PSC should go to the nodes near to the node P. If the time is 10 minutes, the besiege scope should be larger. The smaller scope indicates that the patrol policemen may blockade the nodes that the suspect has passed. Therefore, the besiege program is transferred into the node selection problem in the set of nodes that can be reached within a period of time from the node P. In other words, it is to choose some nodes and determine the corresponding PSCs to blockade the suspect. Upon the road node set and the PSC set, CN is introduced to represent the node set for blockade, whose elements are chosen by (21), where TL denotes the time from when the suspect begins to escape at the node P to the time when the besiege is finished. A primary parameter in BEP is CT. The decision variable $x_{s,n}$ denotes the assignment of the nodes of blockade to PSCs; $x_{s,n} = 1$, when the PSC ($s \in \text{NOD}$) will blockade the node ($n \in CN$); 0, otherwise.

The assignment of PSC to blockaded node should satisfy the time within which the node is reachable (22). In the model proposed, each PSC can only blockade no more than one node (23). At the same time, each node can only be blockaded by no more than one PSC (24):

$$CT_{s,a} + 3 \le CT_{a,P}, \tag{20}$$

$$CN = \{n \in \text{NOD} \mid CT_{n,P} \le TL\}, \tag{21}$$

$$\left(CT_{s,n} + 3\right) \cdot x_{s,n} \le CT_{n,p}, \quad \forall s \in S,\ n \in CN, \tag{22}$$

$$\sum_{n \in CN} x_{s,n} \le 1, \quad \forall s \in S, \tag{23}$$

$$\sum_{s \in S} x_{s,n} \le 1, \quad \forall n \in CN. \tag{24}$$

Based on the above constraints, in order to improve the efficiency of besiege, the patrol-related resources should be utilized fully. In other words, the quantity of the devoted PSCs should be maximized:

$$\max f_1 = \sum_{s \in S, n \in CN} x_{s,n}. \tag{25}$$

The above model does not consider embracement by the blockade node set to the node with contingent accidents. It can be supported by choosing nodes in CN and building assessment criteria for the embracement, as indicated in Algorithm 1. Then, the above model can be applied to optimize the besiege program.

3.4. PSC Assessment Criteria. The quantity of PSCs and the positions of them have global impacts on the efficiency of the patrol-related resources and scheduling plans. The assessment of PSC involves the following aspects. First, from the view of assessed objectives, there are maximal, minimal, and other statistical measures of the quantity of PSCs and the total workloads. Second, the criteria may refer to PSC or to the nodes with alarms. Third, the criteria may measure the service efficiency from the view of PSC or the service satisfaction degree from the view of the residents.

In summary, about the configuration and deployment of PSCs, there are six criteria, which in fact have been indicated in the above studies:

(1) the degree of balance of the total working time weighted by the alarm frequency,

(2) the weighted response time for each node,

(3) the total weighted response time for all nodes,

(4) the total quantity of PSCs,

(5) the blockade time for each main road node,

(6) the total time of blockading all main road nodes.

3.5. Incremental Optimization. The incremental optimization (INO) of the quantities of PSCs is a practical optimization problem. Incremental optimization refers to the fact that the updating program may introduce new PSCs and determine their positions on the basis of the present situation. Obviously, the new set of PSCs should improve performance under the assessment criteria.

Besides the definitions above, the sets, parameters, and decision variables used in INO are defined as follows. The parameter F_n specifies the alarm frequency at the node $n \in \text{NOD}$. The following additional decision variables are introduced; $y_a = 1$, if a PSC is settled at the node $a \in \text{NOD}$; or else 0; $z_{a,e} = 1$, if the main road node $e \in E$ is served by the PSC at the node $a \in \text{NOD}$; or else 0. *sall* represents the total number of PSCs.

Three types of indices are involved in INO: the node set where the alarms are originated and PSCs are positioned; the set of PSCs; the set of main road nodes. Besides the traveling time matrix among road nodes, for the deployment problem, the alarm frequency of node should be considered. Additionally, the referenced service time promised is introduced as a parameter.

The key of PSC incremental optimization is to determine the quantity of new PSCs and their positions under the criteria of PSC deployment. The variable y_a denotes whether a new PSC is settled at the node $a \in N$. Based on the assessment criteria defined above and considering the service time and efficiency, $x_{a,n}$ denotes the assignment of the node $n \in \text{NOD}$ to the PSC settled at the node $a \in \text{NOD}$.

According to the assessment criteria for PSCs, six objectives are defined.

(1) The balance of the total working time weighted by the alarm frequency at node is achieved by minimizing the maximal workload of PSC (26).

(2) The weighted response time for node is to be minimized (27).

(3) The total weighted response time is to be minimized (28).

(4) The total number of PSCs is to be minimized (29).

(5) The longest time to blockade the main road nodes is to be minimized (30).

(6) The total time of blockade is to be minimized (31).

Input $CT_{a,b}$: the traveling time between two nodes $a \in$ NOD and $b \in$ NOD;
$CN \subseteq$ NOD: the initial node set anticipating the blockade;
N: The minimal number of nodes anticipating the blockade;
DIS: the minimal length of time between nodes anticipating blockade.
Output $CN \subseteq$ NOD: the node set anticipating the blockade.
Process
Step 1. Calculate the minimal traveling time among the nodes in CN by: $t = \min_{a,b \in CN} \{CT_{a,b}\}$.
Step 2. The termination criterion of the algorithm:
If $(|CN| \leq N) \vee (t \geq DIS)$ ***Then***
Return
End IF
Step 3. Determine the node to be removed by: $i = \arg\min_{i,a,b \in CN, i \neq a, i \neq b, a \neq b} \{CT_{i,a} + CT_{i,b}\}$.
Step 4. Update the set by: $CN \leftarrow CN \backslash \{i\}$.

Algorithm 1: Node selection for besiege program.

In the following, the constraints of the model INO are formulated ((32)–(48)):

$$\min f_1 = stpma, \tag{26}$$

$$\min f_2 = ntrma, \tag{27}$$

$$\min f_3 = ntrall, \tag{28}$$

$$\min f_4 = sall, \tag{29}$$

$$\min f_5 = etma, \tag{30}$$

$$\min f_6 = etall, \tag{31}$$

$$ntr_n = \sum_{a \in NOD} \left(x_{a,n} \cdot CT_{a,n}\right), \quad \forall n \in \text{NOD}, \tag{32}$$

$$ntr_n \leq MT, \quad \forall n \in \text{NOD}, \tag{33}$$

$$ntrma \geq ntr_n, \quad \forall n \in \text{NOD}, \tag{34}$$

$$stp_a = \sum_{n \in NOD} \left(x_{a,n} \cdot CT_{a,n} \cdot F_n\right), \quad \forall a \in \text{NOD}, \tag{35}$$

$$stpma \geq stp_a, \quad \forall a \in \text{NOD}, \tag{36}$$

$$\sum_{a \in NOD} x_{a,n} = 1, \quad \forall n \in \text{NOD}, \tag{37}$$

$$x_{a,a} = y_a, \quad \forall a \in \text{NOD}, \tag{38}$$

$$ntrall = \sum_{n \in NOD} \left(ntr_n \cdot F_n\right), \tag{39}$$

$$y_a \cdot M \geq \sum_{n \in NOD} x_{a,n}, \quad \forall a \in \text{NOD}, \tag{40}$$

$$y_s = 1, \quad \forall s \in S, \tag{41}$$

$$sall = \sum_{a \in NOD} y_a, \tag{42}$$

$$\sum_{a \in NOD} z_{a,e} = 1, \quad \forall e \in E, \tag{43}$$

$$\sum_{e \in E} z_{a,e} \leq 1, \quad \forall a \in \text{NOD}, \tag{44}$$

$$et_e = \sum_{a \in NOD} \left(z_{a,e} \cdot CT_{a,e}\right), \quad \forall e \in E, \tag{45}$$

$$etma \geq et_e, \quad \forall e \in E, \tag{46}$$

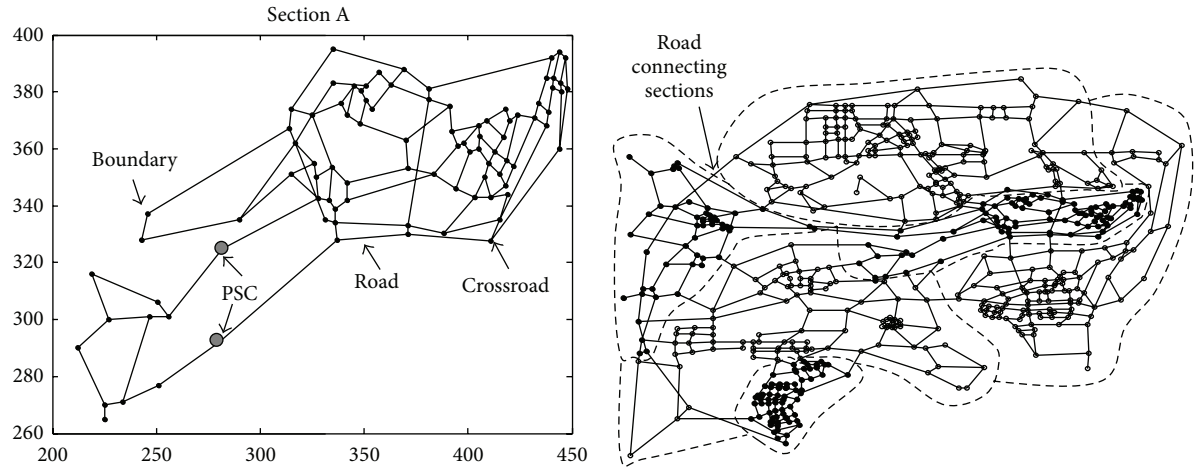
$$etall = \sum_{e \in E} et_e, \tag{47}$$

$$y_a \geq \sum_{e \in E} z_{a,e}, \quad \forall a \in \text{NOD}. \tag{48}$$

Equations (32)–(39) define the basic constraints about the assignment of the nodes with alarms to the PSCs. Equation (32) defines the response time for each node with alarm. Equation (33) sets the restriction of promised service response time to all nodes, where MT is the length of the promised time. This constraint is hard here, which may make the model infeasible. Therefore, it may be removed in the solving process. The maximal response time for any node is defined by (34) based on (32). Equation (35) computes the workload for each PSC. The maximal workload for all PSCs is defined based on (35). By (36), every node with alarm should be assigned to a PSC. Besides (37), (38) defines the constraint that any PSC at least serves the node where it is settled. In (39), the total weighted response time of all nodes is computed.

The constraints ((40)–(42)) primarily deal with the restrictions especially for incremental location problem of PSC. In (40), the assignment variable x is employed to denote the location indicator variable y, where M is a big number whose minimal value is |NOD|. Because it is the incremental location problem, the node which is already a PSC should be PSC in the new solution (41). The number of PSCs after the incremental optimization is computed by (42).

The constraints in (43)–(48) correlate with blockade. Every main road node should be attached to a PSC to blockade it, as defined in (43), whereas (44) sets the rule that each PSC can only take charge of the blockade of one main road node. The time to blockade any main road node is computed by (45), whereas (46) is then used to define the



(a) District A (b) Districts A–F

Figure 2: Traffic network and PSC distribution in sample.

maximal time of blockading any node. Then, the total time of blockade is computed by (47). Of course, only the node where a PSC is settled can be sent to blockade a node (48).

3.6. Redeployment Optimization. The main principle of redeployment optimization (REO) is to employ the original set of PSCs as many as possible. In other words, the difference between the original and new set of PSCs should be minimized. Based on INO (Section 3.5), the original set of PSCs is allowed to change so that the constraint in (41) is removed. New PSC can be added, which requires removing the constraint defined by (42). Then, a new objective should be designed to minimize the quantity of PSCs that are settled at new nodes. In other words, the quantity of new PSCs should be minimized:

$$\min f_7 = \sum_{a \in \mathrm{NOD} \setminus S} y_a. \tag{49}$$

4. Case Study

A network in Figure 2 is used to demonstrate the effects of the above decision models. Because the models built in the study are all ILP or MIP models and the main objective is to build decision models and the general systematic framework for PSC DSS, in the following study, Gurobi 4.6 is employed as a MIP solver to solve the models.

In Figure 2(a), the traffic network of District A with 20 PSCs in a city in China is shown. For the 20 PSCs in District A, there is a coordination problem of how the service areas are probably assigned so that the patrol service can reach the site with contingent event within three minutes (the speed of patrol police car is presumed to be 60 km/h). With the help of the 110 call center and GIS, the position of alarm and the crossroad nearby can be located. The second question is about the emergency blockade to the 13 main crossroads on the boundary of the city by scheduling the 20 PSCs when a contingent event happens. The third question is to determine the quantity and positions of new PSCs to balance the workloads of the current PSCs and shorten the traveling time of patrol services.

Figure 2(b) depicts six sections, A, B, C, D, E, and F. From the view of the assessment and design of all PSCs in the city, the first problem is to assess the present deployment of PSCs. The second problem is what the solution is if it is not rational. The third is the coordination problem among all PSCs in the city. When a serious criminal case occurs at a site in the city, three minutes later the PSCs receive the alarm and the suspect has escaped. To arrest the suspect efficiently, how can the besiege plan be designed and the PSCs be coordinated? This study builds general and extensible models for these problems.

4.1. Service Area Assignment. By GIS, the road network, and PSC deployment data in Figure 2, in the following the above models are demonstrated. In Figure 2(b), there are 582 crossroad nodes, and there are 92 nodes in District A in Figure 2(a). In Table 1, 92 nodes in District A and their frequencies of alarm are listed. The road connection matrix and the distance matrix are ignored in the text. In the city, there are totally 80 PSCs and 20 PSCs in District A. The PSCs and their attached road nodes are listed in Table 2. In Figure 2(b), the quantity of the main nodes on the urban boundary of the city is 17. When a car goes out of the city, it must pass one of these nodes. In Figure 2(a), 13 nodes are such kind of nodes of District A: 12, 14, 16, 21, 22, 23, 24, 28, 29, 30, 38, 48, and 62. In addition, there are totally 928 road segments in the city. By these roads, the shortest distance and path between any two nodes can be calculated by Floyd shortest path algorithm. Then, considering the speed of patrol cars, the time matrix can be computed among all nodes. Because the data volume is too larger, the matrix is not included in the paper. The readers who are interested in the dataset can contact the authors to get it.

The SAA (Sections 3.1–3.3) is multiobjective. In the following, the solution of SAA is computed for District A in

Table 1: Positions and alarm frequencies of nodes in District A.

Number	X	Y	F	Number	X	Y	F	Number	X	Y	F	Number	X	Y	F
1	413	359	1.7	24	212	290	1.1	47	325	372	1.6	70	408	350	0.9
2	403	343	2.1	25	227	300	1.6	48	315	374	1.4	71	415	351	1.1
3	383.5	351	2.2	26	256	301	1.2	49	342	372	1.2	72	418	347	0.8
4	381	377.5	1.7	27	250.5	306	0.8	50	345	382	1.1	73	422	354	0.9
5	339	376	2.1	28	243	328	1.3	51	348.5	380.5	0.8	74	418.5	356	1.1
6	335	383	2.5	29	246	337	1.4	52	351	377	0.6	75	405.5	364.5	0.8
7	317	362	2.4	30	314	367	2.1	53	348	369	1.4	76	405	368	1.1
8	334.5	353.5	2.4	31	315	351	1.6	54	370	363	0.9	77	409	370	0.8
9	333	342	2.1	32	326	355	1.5	55	371	353	1	78	417	364	0.8
10	282	325	1.6	33	327	350	1.4	56	354	374	0.5	79	420	370	0.8
11	247	301	2.6	34	328	342.5	1.7	57	363	382.5	0.8	80	424	372	0.8
12	219	316	2.4	35	336	339	1.4	58	357	387	1.1	81	438	368	1.4
13	225	270	2.2	36	336	334	1.1	59	351	382	0.9	82	438.5	373	1.1
14	280	292	2.5	37	331	335	0.1	60	369	388	0.7	83	434	376	0.9
15	290	335	2.1	38	371	330	1.2	61	335	395	0.6	84	438	385	1
16	337	328	2.6	39	371	333	1.4	62	381	381	1.2	85	440	392	1.2
17	415	335	2.5	40	388.5	330.5	1.7	63	391	375	1.4	86	447	392	1.4
18	432	371	1.9	41	411	327.5	1.4	64	392	366	0.8	87	448	381	1.1
19	418	374	1.8	42	419	344	1.4	65	395	361	0.7	88	444.5	383	0.9
20	444	394	1.9	43	411	343	1.7	66	398	362	0.8	89	441	385	1.4
21	251	277	1.4	44	394	346	1.1	67	401	359	0.8	90	440.5	381.5	0.9
22	234	271	1.4	45	342	342	1.4	68	405	360	0.9	91	445	380	0.9
23	225	265	2.4	46	342	348	1.2	69	410	355	1.1	92	444	360	0.8

Table 2: Attached nodes of PSCs in District A.

Number	Node
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20

Figure 2(a). When the objective functions are ignored in the model, it is found that there is no feasible solution due to the constraint defined by (5). The expectation for minimizing the response time can be achieved by minimizing the objectives f_1 and f_2. Therefore, in the experiments for SAA, the constraint equation (5) is excluded. The priority order of the three objectives is $f_2 > f_3 > f_1$, so that goal programming method is employed in the following steps. First, min f_2 is taken as the objective to minimize the maximal response time. Then, $ntrma = 8.4$ is returned. A new constraint is added to SAA: $ntrma \leq 8.4$. Second, min f_3 is taken as the objective to minimize the total workload. By solving the model, get $ntrall = 150.7$. The following constraint is also then added to SAA as a new constraint: $ntrall \leq 150.7$. Finally, min f_1 is used to balance the workloads. The result is $stpma - stpmi = 14.9$.

Therefore, by simple process on the outputs of the solver, the results (Table 3) are obtained. The total workload and the served nodes by each PSC are presented (Table 3). The rows are sorted by decreasing order of workload, and the workload is normalized after being accumulated (Table 3). The results are then incorporated into Figure 3. The former 6 (30%) PSCs accomplish 50% of the total workload; the former 10 (50%) PSCs finish about 80% workload; and the former 14 (70%) PSCs undertake more than 90% workload (Figure 3). Therefore, the objectives, such as decreasing the alarm response time, enhancing the alarm service quantity, and minimizing the workload, do commonly make the workloads unbalanced between different PSCs. The workloads are mostly served by a few PSCs. Obviously, the quantity of PSCs and their positions impact the service quantity and service efficiency.

Table 3: Total workload and served nodes for every PSC.

PSC	*stp*	Served nodes by PSC
1	9.4	1, 67, 68, 69, 71, 74, 75, 78
2	13.5	2, 38, 40, 43, 44, 70, 72
3	7.7	3, 54, 55, 65, 66
4	8.6	4, 57, 60, 62, 63, 64
5	6.3	5, 49, 50, 52, 53
6	11.4	6, 47, 51, 56, 58, 59
7	12.7	7, 30, 31, 48, 61
8	5.6	8, 32, 33, 46
9	4.4	9, 34, 35, 36, 37
10	0	10
11	2.8	11, 26, 27
12	3.2	12, 25
13	11.3	13, 21, 22, 23, 24
14	0	14
15	14.9	15, 28, 29
16	8.4	16, 39, 45
17	2.8	17, 41, 42
18	10.9	18, 73, 80, 81, 82, 83, 84, 90
19	3.8	19, 76, 77, 79
20	13	20, 85, 86, 87, 88, 89, 91, 92

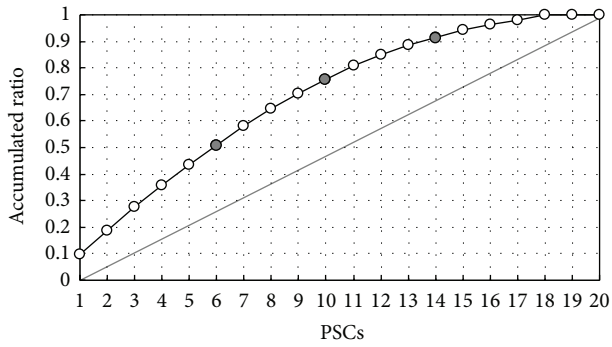


Figure 3: Accumulated workload in decreasing order by PSC.

Table 4: Assignment of PSCs to main road nodes.

Number	Main road node ($e \in E$)	PSC ($s \in S$)	Blockade time (et_e)
1	12	12	0
2	14	16	7
3	16	9	2
4	21	14	4
5	22	10	8
6	23	13	1
7	24	11	4
8	28	15	5
9	29	7	9
10	30	6	4
11	38	2	4
12	48	5	3
13	62	4	1

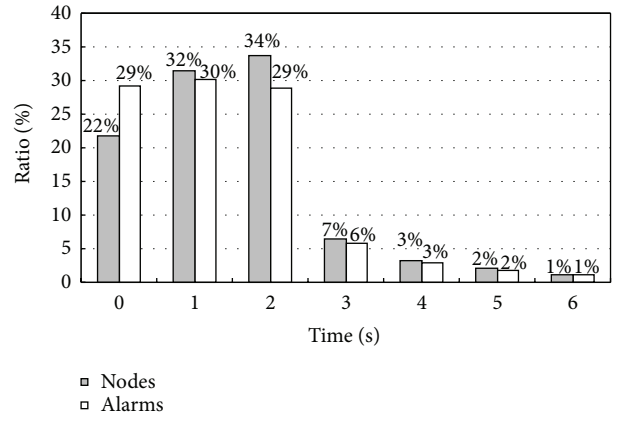


Figure 4: Quantity of nodes and alarms for different length of response time.

In Figure 4, according to the results of solving SAA, the ratio of the quantities of nodes and the ratios of the quantities of alarms are analyzed in the cases of different lengths of response time. When the workloads of PSCs are not balanced, the promise of response within three minutes can be ensured for almost 95% nodes and almost 95% alarms (Figure 4). Therefore, if the patrol-related resource is enough, at least the present deployment can satisfy the service requirements. However, the unbalanced nature indicates certain waste of patrol-related resources (Figure 4).

4.2. Emergent Blockade. Taking District A in Figure 2(a) as a sample, the emergent blockade solution is studied to schedule 20 PCSs to blockade the 13 main road nodes. According to the two objectives in (13) and (14) and the decision variables and constraints, EMB is a multiobjective MIP model. However, when the resources of PSCs are adequate, the key objective is to implement a whole blockade for all main road nodes by minimizing f_1; then the performance of PSC assignment can be further optimized. Therefore, the following steps are conducted to adjust the model and solve it by the dataset of District A (Figure 2(a)). At first, by minimizing the maximal blockade time, min f_1, the result $etma = 9$ and $etall = 76$ is then taken as constraint (50), which is incorporated into the new model. Then, taking min f_2 as the objective to minimize the total blockade time, $etall = 52$ is returned:

$$etma \leq 9. \tag{50}$$

The final assignment solution is then shown in Table 4, where the main road nodes, the PSC to blockade this node, and the blockade time are presented.

4.3. Besiege Program. In BEP, CN can be determined by (21), where CT is a parameter. For example, the node set CN

Table 5: Besiege programs.

7 minutes	8 minutes	9 minutes	10 minutes	15 minutes		
			1 → 40			
		1 → 3	2 → 38	1 → 63	167 → 273	
		2 → 38	3 → 55	2 → 1	168 → 79	373 → 21
	1 → 3	3 → 55	4 → 54	3 → 55	169 → 240	374 → 489
	2 → 38	4 → 54	5 → 5	4 → 54	170 → 216	376 → 22
	3 → 55	5 → 50	6 → 6	5 → 5	171 → 170	383 → 11
3 → 55	4 → 54	6 → 6	10 → 10	6 → 6	172 → 219	384 → 25
5 → 5	5 → 5	10 → 10	15 → 15	10 → 10	173 → 173	475 → 532
6 → 6	6 → 6	15 → 15	16 → 16	11 → 26	174 → 217	476 → 557
15 → 15	10 → 10	16 → 560	17 → 2	12 → 12	175 → 190	477 → 564
16 → 16	15 → 15	17 → 40	18 → 64	14 → 14	176 → 189	478 → 533
171 → 232	16 → 16	19 → 44	19 → 44	15 → 15	178 → 220	480 → 551
173 → 173	171 → 231	168 → 60	168 → 60	16 → 16	179 → 222	481 → 531
	172 → 244	170 → 171	170 → 171	17 → 64	180 → 223	482 → 475
	173 → 173	171 → 232	171 → 229	18 → 62	181 → 274	484 → 529
		172 → 231	172 → 228	19 → 65	182 → 214	485 → 480
		173 → 173	173 → 173	20 → 69	372 → 27	
			174 → 216			

within $CT = 10$ minutes' distance from node 32 contains the nodes as shown in the following:

$$CN = \left\{ \begin{array}{c} 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, \\ 16, 28, 30, 31, 33, 34, 35, 36, 37, 38, \\ 39, 40, 44, 45, 46, 47, 48, 49, 50, 51, \\ 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, \\ 63, 64, 65, 66, 67, 171, 173, 216, 228, 229, 230 \end{array} \right\}. \tag{51}$$

Setting TL to 7, 8, 9, 10, and 15, the besiege programs by BEP can be obtained (Table 5), where "$a \rightarrow b$" denotes "the node of PSC → the blockaded node," which represents the assignment of PSC to the blockaded node.

4.4. Incremental Optimization. The model INO is a multiobjective model with six objectives. This study does not aim at a general algorithm for multiobjective MIP model. In the following, the steps considering the priorities of all objectives are designed to solve the model.

(1) Due to the high cost of building PSC, the quantity of PSCs should be minimized. By taking min f_4 as objective, $sall = 24$ will be returned, while the alarms from all nodes can be served within 3 minutes. Therefore, a new constraint as (52) is defined for the new model to limit the quantities of PSCs.

(2) Taking min f_5 as objective to minimize the longest blockade time for serious contingent accident, $etma = 8$ is returned. Therefore, a new constraint in (53) is incorporated.

(3) Taking min f_6 as objective to minimize the total blockade time, $etall = 33$ is returned. Similarly, a new constraint as (54) is obtained.

(4) Taking min f_2 as objective to minimize the longest weighted response time, $ntrma = 4.8$ is returned. A new constraint is defined by (55).

(5) Then, taking min f_1 as objective to balance the workloads between different PSCs, $stpma = 11.3$ is returned and a new constraint is (56).

(6) Finally, taking min f_3 as objective to minimize the weighted response time, $ntrall = 117.1$ is returned, and the final results are shown in Tables 10 and 11.

In Table 6, the PSC, its attached node, its workload, and served nodes are presented. The notation "new" indicates that the PSC is a new one suggested by solving the model INO. On the basis of the original 20 PSCs, four new PSCs are settled at the nodes 29, 38, 48, and 89.

In Table 7, the assigned PSC and the time for blockade are shown when all 13 main road nodes are required to be blockaded.

By removing the constraint defined by (43), INO is then changed to a model that directly locates all PSCs. Moreover, by incorporating the quantity limit of PSCs, the new results (Table 8) can be obtained by similar steps proposed above. Besides the solution of 24 PSCs after incremental optimization, two additional solutions with 15 and 16 PSCs are obtained (Table 8). Different quantity of PSCs will produce different values of various criteria (Table 8). However, apparently the solution with more PSCs will show better performances of almost all criteria:

$$sall \leq 24, \tag{52}$$

$$etma \leq 8, \tag{53}$$

$$etall \leq 33, \tag{54}$$

$$ntrma \leq 4.8, \tag{55}$$

$$stpma \leq 11.3. \tag{56}$$

4.5. Assessment of the Present PSCs. For the dataset of the entire city with 582 nodes and 80 PSCs, by adjusting the

Table 6: PSC, its workload, and served nodes.

PSC	Workload	Nodes
1	11	1, 67, 68, 69, 71, 72, 74, 75, 78
2	7.1	2, 40, 43, 44, 70
3	7.7	3, 54, 55, 65, 66
4	8.6	4, 57, 60, 62, 63, 64
5	9.4	5, 49, 50, 51, 52, 53, 56
6	5.1	6, 58, 59
7	8.3	7, 30, 32, 47
8	2.6	8, 33, 46
9	7	9, 34, 35, 36, 45
10	0	10
11	2.8	11, 26, 27
12	3.2	12, 25
13	11.3	13, 21, 22, 23, 24
14	0	14
15	4.8	15, 31
16	0.2	16, 37
17	2.8	17, 41, 42
18	7.1	18, 73, 80, 81, 82, 83
19	3.8	19, 76, 77, 79
20	2.6	20, 85, 86
29 (new)	1.3	28, 29
38 (new)	1.4	38, 39
48 (new)	1.8	48, 61
89 (new)	7.2	84, 87, 88, 89, 90, 91, 92

Table 7: Main road node, time to blockade it, and the PSC that blockades it.

Main road node	Blockade time	PSC
12	0	12
14	7	16
16	2	9
21	4	14
22	8	10
23	1	13
24	4	11
28	5	15
29	0	29
30	1	7
38	0	38
48	0	48
62	1	4

model in Section 3.4, the assessment is conducted. At first, the original setting of the 80 PSCs is accepted (43). Then, (57) is added to restrict the entrance of new PSCs. Further, the promised response time is adjusted to $MT = 30$.

The results (Table 9) present the assessment result of the current PSC configuration and deployment. The values about "blockade time," "response time," and "workload" are a bit bigger than the values in Table 8. The present solution can be optimized further.

By incremental optimization and direct optimization, new configuration and deployment programs for PSC are studied. The solutions are assessed under the six criteria:

$$y_s = 0, \quad \forall s \in \mathrm{NOD} \setminus S. \tag{57}$$

4.6. Incremental Optimization and Assessment. The first method to improve the PSC system is to add new PSCs. In the following, new PSCs are introduced to satisfy the premised service time. First, (43) is accepted and (57) is removed, so that the model is changed to INO. Second, MT is set to 3 and 5. Then, two solutions are returned with the assessment result (Table 10). When $MT = 3$, 134 PSCs are required. It costs much because 64 PSCs should be added. However, the assessment result of other criteria all shows that it is really good. When $MT = 5$, the quantity of PSCs decreases to 94 and the assessment values are deteriorated rationally.

4.7. Direct Optimization and Assessment. Direct optimization uses a revised version of INO where (43) and (57) are removed. Similarly, according to the steps proposed above, when MT is set to 3, 4, and 5, the assessment result is obtained (Table 11). The solutions cannot be compared directly because multiple criteria are involved. Moreover, the solving strategies also affect the results greatly.

4.8. Redeployment Optimization. When $MT = 3$, get $sall = 94$. Then, by the revised model above, a new set of PSCs can be obtained. By solving the model, $f_7 = 14$ is returned while the original PSCs are all kept in the new solution. When $sall = 94$, if it is expected to minimize $ntrall$, the result is $f_7 = 60$. Therefore, only 34 original PSCs are kept in the new solution.

5. Conclusion

The quantity of patrol service centers and their locations and the scheduling of patrol-related resources are important for urban security and protection. A general framework of PSC DSS and a series of decision models are proposed, which are demonstrated in a city with six districts. From the view of optimizing and scheduling the patrol-related resources, the assessment criteria for PSC deployment are designed, and six models are built. The PSC service area assignment, emergent blockade, and emergent besiege implement the solutions of assignment of PSCs to alarms and dynamic organization of PSCs when contingent events happen. The independent service by a single PSC and the coordinated response by multiple PSCs are both focused on. Based on the incremental optimization model, the general assessment and optimization models are built. The results of the strategies including direct optimization, incremental optimization, and redeployment are analyzed. As an important contribution, the study developed a systematic and holistic integrated solution for the deployment and various important scheduling scenarios of PSCs.

The study involves a wide scope of the models for the operations and management of urban PSCs, whereas only basic deterministic models are examined here. During the

Table 8: Performance comparison of incremental and direct optimization for PSCs.

Criterion	*sall*	*etma*	*Etall*	*ntrma*	*stpma*	*ntrall*
Incremental	24	8	33	4.8	11.3	117.1
Direct	15	8	45	7.5	27.6	199.3
Direct	16	8	24	5.2	19.1	185.1

Table 9: Assessment of the PSCs.

Criterion	*sall*	*etma*	*etall*	*ntrma*	*stpma*	*ntrall*
Current situation	80	13	93	14.4	82	1367.3

Table 10: Incremental optimization for the PSCs.

Criterion	*MT*	*sall*	*etma*	*etall*	*ntrma*	*stpma*	*ntrall*
Incremental	3	134	4	8	4.8	34	818.4
Incremental	5	94	5	29	8.5	62	1168

Table 11: Direct optimization of the configuration of PSCs.

Criterion	*MT*	*sall*	*etma*	*etall*	*ntrma*	*stpma*	*ntrall*
Direct	3	94	63	550	8	56	1316
Direct	4	63	50	533	11	95	1744
Direct	5	45	66	520	14	170	2160

process of the research, some related subjects are discovered and they warrant further studies. First, in the aspects of determining the quantity and locations of PSCs, the service demands originated by averaged quantities of alarms cannot reflect the uncertainty of the demands. Therefore, the stochastic optimization can be a new direction. Second, the practical PSCs will have different capability and capacity for patrol-related services, which impose additional complexities. Third, the parallel service capability of single PSC is not indicated in the study. In practice, a PSC can deal with many alarms at the same time. During blockade and besiege, a PSC can take charge of multiple nodes. Fourth, the starting position for a service may not be the settled node of the PSC. The patrol policeman can accept demands during patrol processes. Therefore, dynamic routing and real-time scheduling models are topics of practical significance.

Conflict of Interests

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

Acknowledgments

This study is partially supported by the National Nature Science of China (71101088, 71171129, 71390521, and 71471109), the Science Foundation of Ministry of Education of China and Shanghai (20113121120002, 14YZ100, 20123121110004, and 13SG48), and the Science and Technology Commission of Shanghai (14DZ2280200 and 12510501600).

References

[1] D. Reis, A. Melo, A. L. V. Coelho, and V. Furtado, "Towards optimal police patrol routes with genetic algorithms," in *Intelligence and Security Informatics*, vol. 3975 of *Lecture Notes in Computer Science*, pp. 485–491, Springer, Berlin, Germany, 2006.

[2] R. T. Coupe and L. Blake, "The effects of patrol workloads and response strength on arrests at burglary emergencies," *Journal of Criminal Justice*, vol. 33, no. 3, pp. 239–255, 2005.

[3] S. J. D'Amico, S.-J. Wang, R. Batta, and C. M. Rump, "A simulated annealing approach to police district design," *Computers and Operations Research*, vol. 29, no. 6, pp. 667–684, 2002.

[4] H. C. W. Lau, G. T. S. Ho, Y. Zhao, and W. T. Hon, "Optimizing patrol force deployment using a genetic algorithm," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8148–8154, 2010.

[5] Y. Lou, Y. Yin, and S. Lawphongpanich, "Freeway service patrol deployment planning for incident management and congestion mitigation," *Transportation Research C: Emerging Technologies*, vol. 19, no. 2, pp. 283–295, 2011.

[6] B. B. Keskin, S. R. Li, D. Steil, and S. Spiller, "Analysis of an integrated maximum covering and patrol routing problem," *Transportation Research E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 215–232, 2012.

[7] Y. Yin, "Optimal fleet allocation of freeway service patrols," *Networks and Spatial Economics*, vol. 6, no. 3-4, pp. 221–234, 2006.

[8] Y. Yin, "A scenario-based model for fleet allocation of freeway service patrols," *Networks and Spatial Economics*, vol. 8, no. 4, pp. 407–417, 2008.

[9] R. Pal and K. C. Sinha, "Simulation model for evaluating and improving effectiveness of freeway service patrol programs," *Journal of Transportation Engineering*, vol. 128, no. 4, pp. 355–365, 2002.

[10] S. E. Bodily, "Police sector design incorporating preferences of interest groups for equality and efficiency," *Management Science*, vol. 24, no. 12, pp. 1301–1313, 1978.

[11] J. M. Carroll and P. G. Laurin, "Using simulation to assign police patrol zones," *Simulation*, vol. 36, no. 1, pp. 1–12, 1981.

[12] G. M. Kern, "Computer simulation model for the study of police patrol deployment," *Simulation*, vol. 52, no. 6, pp. 226–232, 1989.

[13] K. M. Curtin, K. Hayslett-McCall, and F. Qiu, "Determining optimal police patrol areas with maximal covering and backup covering location models," *Networks and Spatial Economics*, vol. 10, no. 1, pp. 125–145, 2010.

[14] T. Assunção and V. Furtado, "A heuristic method for balanced graph partitioning: an application for the demarcation of preventive police patrol areas," in *Advances in Artificial Intelligence—IBERAMIA 2008*, vol. 5290 of *Lecture Notes in Computer Science*, pp. 62–72, Springer, Berlin, Germany, 2008.

[15] A. Khattak, N. Rouphail, K. Monast, and J. Havel, "Method for priority-ranking and expanding freeway service patrols," *Transportation Research Record*, vol. 1867, pp. 1–10, 2004.

[16] J. M. Chaiken and P. Dormont, "A patrol car allocation model: background," *Management Science*, vol. 24, no. 12, pp. 1280–1290, 1978.

[17] J. M. Chaiken and P. Dormont, "A patrol car allocation model: capabilities and algorithms," *Management Science*, vol. 24, no. 12, pp. 1291–1300, 1978.

[18] L. C. Green and P. J. Kolesar, "A comparison of the multiple dispatch and M/M/c priority queueing models of police patrol," *Management Science*, vol. 30, no. 6, pp. 665–670, 1984.

[19] L. V. Green and P. J. Kolesar, "Testing the validity of a queueing model of police patrol," *Management Science*, vol. 35, no. 2, pp. 127–148, 1989.

[20] S. R. Sacks, "Optimal spatial deployment of police patrol cars," *Social Science Computer Review*, vol. 18, no. 1, pp. 40–45, 2000.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 723463, 13 pages
http://dx.doi.org/10.1155/2015/723463

Research Article

Parallel Control to Fragments of a Cylindrical Structure Driven by Explosive inside

Wenkai Chen, Xiangyu Li, Fangyun Lu, Zhenduo Li, and Zhenyu Zhang

College of Science, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Xiangyu Li; xiangyu_lee@163.com

Received 30 August 2014; Accepted 8 December 2014

Academic Editor: Yi-Chung Hu

By analyzing the fragmentation distributions of a cylindrical structure and a specific structure, the necessity of parallel control to the fragments is presented. The shell shape of structures has an influence on the fragment spatial distribution. A new design method for the shell shape is proposed. To facilitate the establishment of the numerical model and the machining for relative experiments, the mathematical description of the theoretical calculated generatrix of the shell is simplified. The fragment spraying processes of the designed structures are simulated, and end effects are analyzed. Based on the theoretical design and plentiful simulation data, the relationships between the size of the parallel fragmentation structure and the optimized curvature radius of the shell are expressed by an equation. The equation is validated by numerical means and can be a reliable reference to the design of the parallel fragmentation structure.

1. Introduction

Cylindrical structure with explosive inside is widely used in the mining, blasting engineering, and military weapon. In modern warfare, the vulnerability of the target is greatly reduced with the development of technology. Due to the enhanced protection of every component, the target is often still able to work, when it suffers a large area of low-density damage. However, the dependence of the target on every component rises. Sometimes only a component failure can lead to the complete strike of the target [1, 2]. Consequently, the most efficient path to destroy the target is the high-density regional damage [3, 4].

For the conventional cylindrical structure, because of the end effects caused by the rarefaction wave at the ends of the structure, the ejection angle of fragments increases. That is to say, the width of the fragments spatial distribution along the axial direction becomes wider, which lowers the density of fragment distribution. Thus, the fragments could not achieve a high-density damage to the target and become ineffective for the most parts. The further the fragments spray, the lower the utilization ratio of the fragments is.

Modern requirements for increased warhead lethality with decreased mass, together with their use in complex systems, mean that many systems are amiable [5, 6]. Focusing fragmentation warhead through the specific structure decreases the ejection angle of fragments. The fragments are gathered together in a much narrower spatial width, which makes a cutting damage to the target come true. However, owing to the immutability of the specific structure, the high ability of focusing fragments can only be achieved in a constant distance, and fragments are likely to collide with each other in the focusing region [7–9]. In the actual encountering situation between the missile and the target, the missile and the target usually occur at a high speed impact, which would lead to large differences between the actual encountering distance and the desired distance. The large difference in the encountering distance leads to the reduction of damage effect to the target. The fragments of structures are always densely arranged in the assembly process, which makes the fragment density maximize along the axial direction optimally. So the best way to achieve the ability to focus fragments along the axial direction is by making all the fragments spray horizontally, perpendicular to the axial direction.

The parallel fragmentation structure presented in this paper is based on the design of the focusing fragmentation warhead and can achieve high-density effective damage to

Figure 1: Facade of structure.

the target. The key technology of the structure is the design of the shell shape. This paper demonstrates a new method to design the shell shape of the structure and a new definition of deflecting angle, which is a new parameter to describe the performance of fragments. The equation that describes the relationship between the size of the parallel fragmentation structure and the curvature radius of the arc equalizing the generatrix of shell is attained and optimized with the relevant numerical simulation results. The equation can be a quick and easy design method for the parallel fragmentation structure.

2. Experiment

In the experiments related to conventional cylindrical structures, it is easy to find that the fragments distribute dispersedly and the width of the fragments' spatial distribution along the axial direction is much bigger than the height of the structure.

2.1. Experiment Preparation. The cylindrical structure is centrally symmetrical and the shape is cylindrical, as shown in Figure 1. The outer diameter of the structure is 180 mm and the height is 200 mm. The cylindrical structure in the experiment comes in several parts, as seen in Figure 2. The covers are made of 10 mm thick LY-12 aluminum, and the shells are made of 2 mm thick 1020 steel. The fragments are alloy steel spheres, and the diameter of the spheres is 8 mm. All the fragments are arranged closely to each other and fixed with epoxy resin between the inner shell and the outer shell. The explosive is JHL-3, which is a type of high energy explosive, and is detonated by an 8 mm diameter exploder from the top of structure [10–12].

The target is 3 mm thick, 1.5 m long, and 1.26 m wide steel plane. To simplify the collection and analysis of experimental data, we mesh the target into 20 lattices, as depicted in Figure 3.

Figure 4 shows the arrangement of target and the structure. The distance between the structure and target is 3.5 m. The structure is arranged vertically on a wooden frame, and the axis of the structure is parrallel to the ground. After the explosive is detonated, the shock wave appears and reflects from the ground to the original position. In order to avoid the effects of the shock wave on the process that fragments

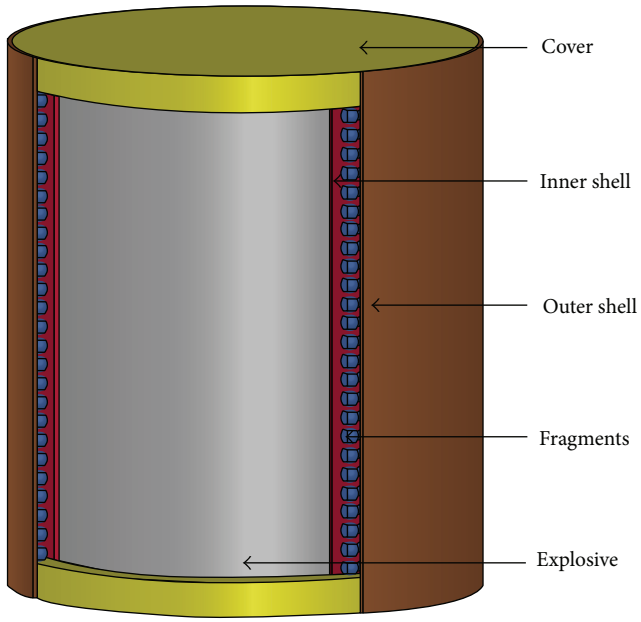


Figure 2: Components of cylindrical structure.

Figure 3: Target plane.

spray away from shell, the distance between the structure and the ground should be not less than 1.5 m [14]. On the circle whose center is the structure and radius is 3.5 m, two targets are arranged. Each target can blanket 20.74° of the circle, so 11.52% of the fragments can be intercepted by the two targets. The two targets and the structure are in a line. Besides, the equatorial plane of the structure crosses the line which is 0.8 m far away from the upper edge of the target.

2.2. Damage to the Targets. Because the structure is centrally symmetrical, the distribution of fragments on the two targets can reflect the characteristics of all the fragments adequately. Figure 5 shows the damage to the targets made by the structure, and we can observe that there are plenty of holes on the targets. A lot of big holes are made by bursting the shells in high velocity, whilst the majority of the regular holes

Figure 4: Arrangement of target and the structure.

Figure 5: Damage to the targets by the structure.

are made by the fragments. Based on the distribution of the holes on the targets, we calculate a number of statistics about the number of fragments crossing the targets, as depicted in Figure 6.

2.3. Discussion. Firstly, we can see that the region of fragments nearly covers all the targets. The width of fragment distribution along the axial direction is wide. With the increase of spraying distance, the width would become wider and many more fragments would become invalid.

Secondly, it is obvious that the whole distribution of fragments is slightly to the downside of the equatorial plane of the structure. The main influences are that the explosive is detonated from the top of the structure and the shock wave propagates from up to down. The direction where the shock wave propagates makes a significant impact on the direction where the fragments spray.

Thirdly, it seems that part of the fragments concentrate in the range from 30 cm to 60 cm on the targets. In fact the spraying directions of fragments near the ends deviate from the equatorial plane of the structure too much, which is caused by the end effects.

Optimizing the distribution of fragments, the structure is redesigned. This paper presents a new cylindrical structure, named as the parallel fragmentation structure.

3. Analysis and Structure Design

The significant characteristics of the parallel fragmentation structure are that the fragments spray in parallel. In this

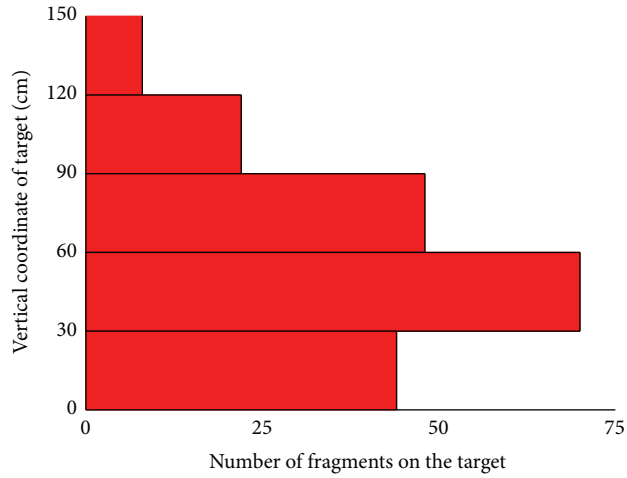


Figure 6: Statistics about the number of fragments crossing the targets.

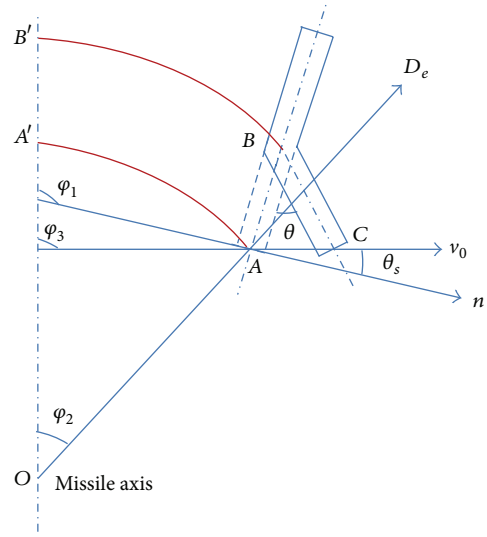


Figure 7: Calculation of ejection angle of fragments.

section, the details on the spraying process of fragments are discussed, followed by a method to design the shape of the shell which is proposed, and, lastly, the case of end effects is considered.

3.1. Process of Fragments Spraying. According to the Shapiro theory, the detonation wave propagates in the form of a spherical wavefront, outward from the booster explosive or the booster detonator.

Now let us take an infinitesimal element AB of the shell as the study point [15, 16]. In Figure 7, φ_1, φ_2, and φ_3 are the angles between the missile axis and the normal line of the shell, the normal line of detonation wavefront, and the velocity vector of the infinitesimal element AB, respectively. θ_s is the angle between the normal line of the shell and the velocity vector of the infinitesimal element AB. θ is the angle that the infinitesimal element AB turns. O is the detonation point.

Here, we put forward a series of hypotheses about the detonation wave. The detonation wave propagates to the ends from the center of the structure. The direction of the detonation wave keeps unaltered, after the detonation wave impacts the shell of the structure.

The law of fragments spraying away from the shell is fitted to Shapiro law, expressed as

$$\tan\theta_s = \frac{v_0}{2D_e}\cos\left(\frac{\pi}{2} - \varphi_1 + \varphi_2\right), \tag{1}$$

where v_0 is the velocity of fragment spraying from the shell and D_e is the explosive velocity. We can see that the ejection angle of fragments is related to the detonation point and the direction where the detonation wave propagates.

The Shapiro law indicates that the time for the infinitesimal element AB to accelerate to its final speed from rest is instantaneous. During the derivation of Shapiro law [17], the distance between point A and point C is expressed as

$$\overline{AC} = v_0\Delta t. \tag{2}$$

Δt is the period of time for the element AB transforming to CB. However, the acceleration of the infinitesimal element AB cannot be neglected. Because the infinitesimal element keeps accelerating throughout the expanding process, we can regard $v_0/2$ as the average velocity during the process where the point A of element AB transforms to point C. The relationship (2) is redefined as

$$\overline{AC} = \frac{1}{2}v_0\Delta t. \tag{3}$$

Based on the above analysis, the Shapiro law (1) is revised as

$$\tan\theta_s = \frac{v_0}{4D_e}\cos\left(\frac{\pi}{2} - \varphi_1 + \varphi_2\right). \tag{4}$$

From Figure 7, we know the relationship among φ_1, φ_3, and θ_s to be

$$\varphi_1 = \theta_s + \varphi_3. \tag{5}$$

The most attractive characteristic of the new structure, proposed in this paper, is that the fragments spray in parallel; that is, the angle φ_3 is 90 degrees. The relationship (5) can be written as

$$\varphi_1 = \theta_s + \frac{\pi}{2}. \tag{6}$$

Using the above relationship (6) between φ_1 and θ_s, the revised Shapiro law (4) can be simplified as

$$\tan\theta_s = \frac{v_0}{4D_e}\cos\left(\theta_s + \varphi_2\right). \tag{7}$$

The parameter φ_2 can be calculated through the initial located position of the infinitesimal element and the initiation point. The explosive velocity D_e is determined by the type of explosive. With the solution of v_0 in the next section, the angle θ_s also can be solved.

Table 1: Parameters of explosives [13].

Explosive category	D_e (m·s^{-1})	ρ_e (g·cm^{-3})	$\sqrt{2E}$ (m·s^{-1})
TNT	6640	1.59	2316
Comp.B	7840	1.68	2682
RDX	8180	1.65	2834
HMX	8600	1.84	2895
PETN	7980	1.73	2834
Tetry	7850	1.62	2500
Octol	8643	1.80	2895

3.2. Initial Fragment Velocity and Gurney Specific Energy. The initial fragment velocity v_0 is expressed in the Gurney formula [18–20], which is given by

$$v_0 = \sqrt{2E}\sqrt{\frac{\beta}{1 + 0.5\beta}}, \tag{8}$$

where β is the ratio of charge mass to the shell mass of the unit height cylinder and $\sqrt{2E}$ is the Gurney constant or Gurney specific energy. Every explosive has a specific Gurney constant, which represents the component and feature of explosive, as shown in Table 1. The Gurney constant is related to the loading density of the explosive and other factors. By experimentation, a linear correlation is found between the Gurney constant and the explosive velocity and is given by

$$\sqrt{2E} = 0.52 + 0.25D_e, \tag{9}$$

where the explosive velocity (D_e) and the Gurney constant ($\sqrt{2E}$) have the same dimension.

3.3. Structure Design. The finite method is used for the design of structure in any size. The design of the parallel fragmentation structure is shown in Figure 8. The origin of the coordinate system is located at the geometrical center of the structure, and the y-axis with respect to the coordinate is defined by the space-axial position. In Figure 8, the dotted line is a quarter boundary of the half-section of the shell, and the length of the infinitesimal element (M_1M_2) is l.

According to the height (H) and the diameter (L) of the structure, we can get the coordinate of point M_1 ($H/2$, $L/2$). Then, by means of a trigonometric calculation, the value of angle φ_2 is obtained. Inserting the value of angle φ_2 to (5), the numerical solution of the angle (θ_s) between the normal line of the shell and the velocity vector of the infinitesimal element (M_1M_2) can be solved, with the results that the coordinate of M_2 can be given by ($X_{M_1} - l*\sin\theta_s, Y_{M_1} - l*\cos\theta_s$). By analogy, we can get every terminal coordinate of the infinitesimal elements, such as $M_3, M_4, \ldots, M_n$, depicted in Figure 8. Liner fitting to all terminals can lead to an equation of the generatrix of the shell. In order to facilitate the establishment of the numerical model and the machining for relative experiments, we equate the generatrix of the shell to an approximate arc, whose curvature radius can be calculated by the relevant geometric relationships as shown in Figure 9.

For a designed fragmentation structure whose height and diameter are both 200 mm, the comparison of the theoretical

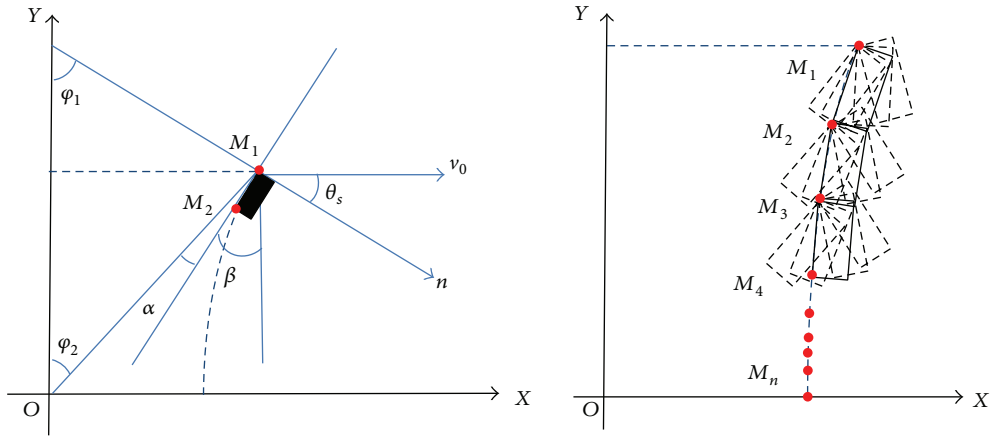


Figure 8: Design approach.

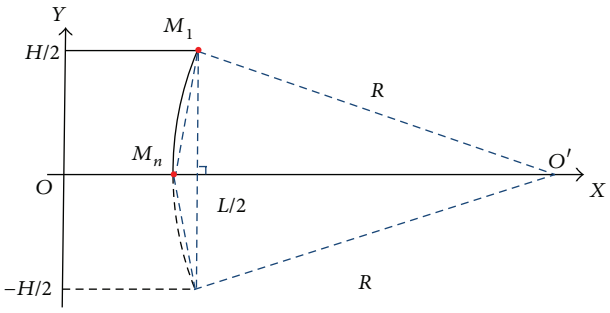


Figure 9: Geometric relationships to calculate curvature radius.

calculated generatrix of its shell and the equivalent arc is shown in Figure 10. It should also be noted that the coordinate system in Figure 10 is similar to the one in Figure 8. It is obvious that, in general, the equivalent arc is similar to the theoretical calculated generatrix and simplifies the mathematical description of the generatrix effectively.

3.4. End Effects. One of the problems in the design of structures is the end effects influencing the ejection angle of fragments near the ends. It is well known that significant differences exist between the ejection angle of fragments located in the center of the structure and those of fragments located near the ends of the structure. For the purpose of minimizing the end effects, a new component of platform is proposed [21]. Next, we will talk about the new definition of the deflecting angle proposed in this paper.

The performance of fragmentation structures is usually described with characteristics of the fragment dispersion and includes mass, ejection angle, direction angle, velocity, and the distribution density of fragments [22–25]. The ejection angle (φ) and direction angle (Ω), shown in Figure 11, are studied widely in designing fragmentation structures. The direction angle is the angle between the midline with 45% effective fragments on each side and the equatorial plane through the barycenter. All the explosives are detonated from the geometrical center of the structures, and the parallel fragmentation structure is symmetrical, so the direction angle fluctuates around 0° and is not discussed further in this paper. The ejection angle is the cone angle which includes 90% effective fragments.

However, the ejection angle cannot describe the relationship between the fragment dispersion and the structure but can only show the characteristics of the fragment dispersion. The deflecting angle (θ) is proposed in this paper, which is the average between the upper deflecting angle (θ_1) and the lower deflecting angle (θ_2). The upper deflecting angle (θ_1) is the angle between the upper boundary of the ejection angle and the plane through the upper surface of the structure, so as to the lower deflecting angle (θ_2).

The fragment spraying process of the cylindrical structure, whose height and diameter are both 200 mm, and the modified structure with 30 mm high platform on each end is simulated. The fragment distributions on the target plane (distance 3.5 meters) are shown in Figure 12. It is obvious that the platform can minimize the end effects effectively. The deflecting angle of the cylindrical structure is 8.42°. The angle of the modified cylindrical structure is 3.96°, reducing the reflecting angle by 4.46°. The modified structure is 60 mm longer than the cylindrical structure. Considering whether the change of structure height makes the decrease of deflecting angle, another cylindrical structure is modeled, which is of 260 mm height and 200 mm diameter. The deflecting angle of the longer cylindrical structure is 8.85°, which is similar to the one of 200 mm height cylindrical structure, so the height of the structure does not play an important role in influencing the deflecting angle and the platform can minimize the end effects effectively.

4. Numerical Simulation and Discussions

The numerical model is made using the nonlinear, explicit finite element code LS-DYNA (version 970). The method to design the shell shape is tested by numerical simulation. Considering the influences on the deflecting angle, the parallel fragmentation structure is optimized.

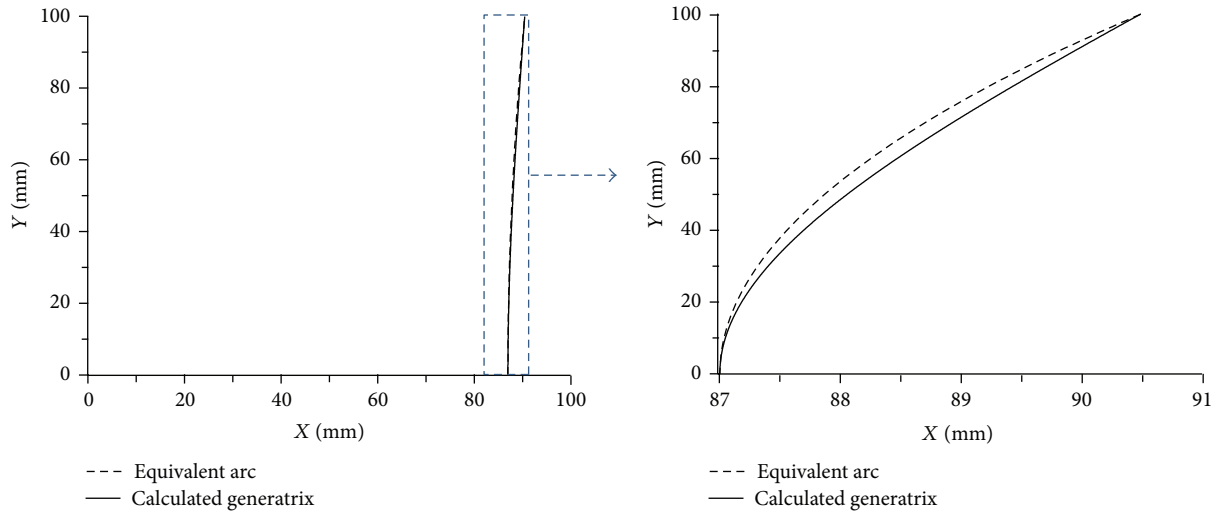


Figure 10: Comparison of the theoretical calculated generatrix of the inner shell and the equivalent arc.

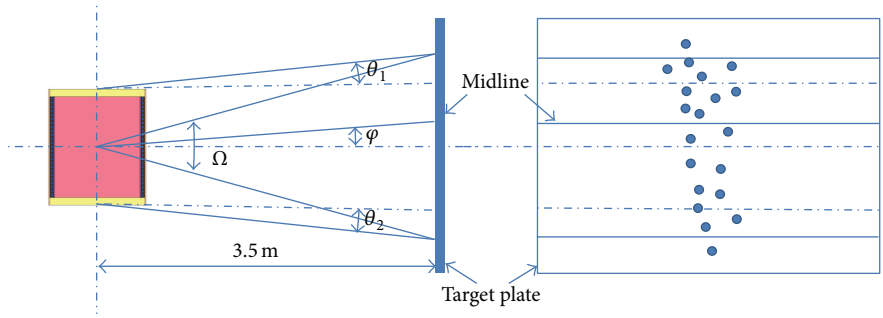
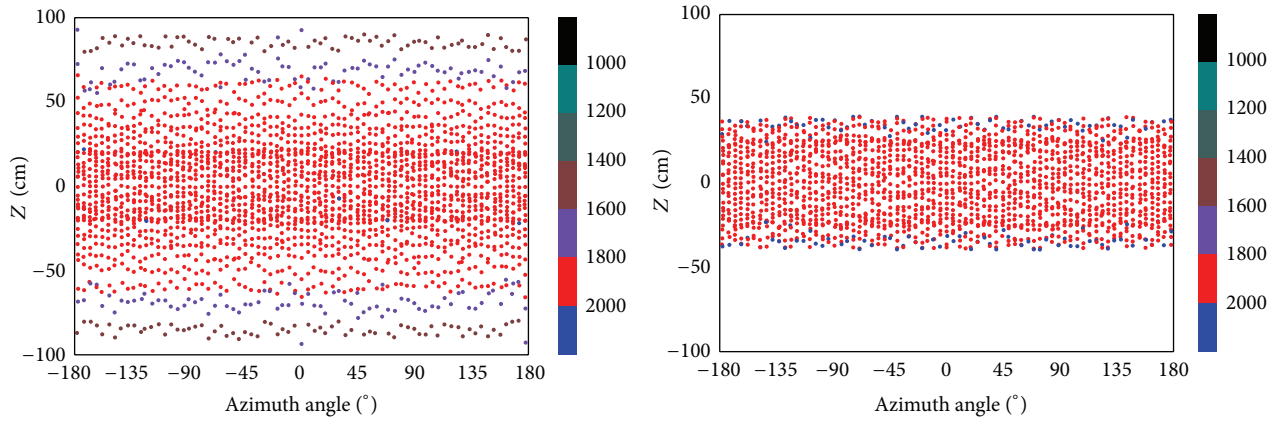


Figure 11: Ejection angle (Ω), direction angle (φ), and deflecting angle (θ).



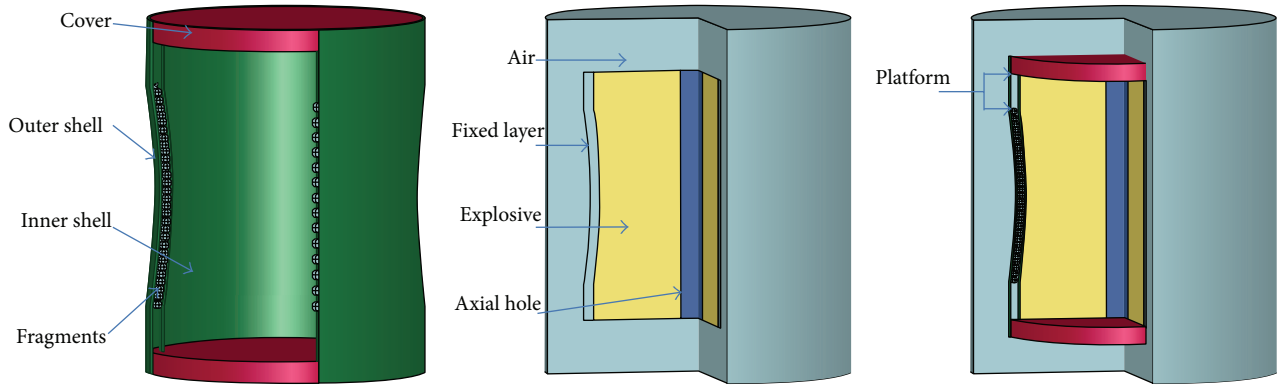
(a) Cylindrical structure

(b) Modified structure with platform

Figure 12: Fragment distributions on the target plant (distance 3.5 meters).

4.1. Model and Material. The numerical model of the parallel fragmentation structure is developed with the nonlinear, explicit finite element code LS-DYNA. This software is particularly suitable for nonlinear dynamic problems, such as impact or explosion. It allows for the application of different algorithms such as Euler, Lagrange, arbitrary Lagrange-Euler (ALE), and SPH to solve the fluid-structure problems [26–28].

Eight-node brick elements with ALE were adopted for the structure. Since the ALE approach uses meshes that are embedded in material and deform with the material, it combines the best features of both Lagrange and Euler



(a) Lag (b) Euler (c) Whole

Figure 13: Numerical model.

methods and allows the mesh within any material region to be continuously adjusted in arbitrary and predefined ways as the calculations proceed. Therefore, the ALE approach is suitable to analyze the process of explosion and fragments spraying away from the shell.

4.1.1. Numerical Model. As shown in Figure 13, the numerical model of a parallel fragmentation structure is separated into Lagrange model and Euler model, including cover, outer shell, inner shell, fragments, air, fixed layer, explosive, and axial hole. The fragments are arranged in the fixed layer between the inner shell and the outer shell, filled with epoxy resin. The platform is used to minimize the end effects. If the explosive is detonated from the geometrical center of the structure in the experiments, the axial hole should exist in the middle of the structure for the exploder. In order to be more consistent for experiments, an axial hole full of sponge is considered in the numerical model and the diameter of the hole is 32 mm.

According to the various functions of the structure, the magnitude of the structure differs, as do the size of fragments and the thickness of the shell. The parallel fragmentation structure designed in this paper is suitable for airborne missiles. In the simulations, the height and the diameter of the structure range from 150 mm to 300 mm. All the fragments are spheres, whose diameters are 6 mm. The thickness of the shells is 1.5 mm.

4.1.2. Modeling Material. A high explosive material model (*MAT_HIGH_EXPLOSIVE_BURN) incorporating the JWL equation of state (EOS_JWL) was used to describe the material property of the RDX charge. The EOS_JWL equation defines the pressure as a function of relative volume and internal energy per initial volume; that is,

$$p = A\left(1 - \frac{\omega}{R_1 V}\right)e^{-R_1 V} + B\left(1 - \frac{\omega}{R_2 V}\right)e^{-R_2 V} + \frac{\omega E}{V}, \quad (10)$$

where p is the blast pressure, E is the internal energy per initial volume, V is the initial relative volume, and ω, A, B, R_1, and R_2 are material constants, respectively.

The key material types and material properties of cover, shell, fragment, and RDX charge, as well as the parameters of equations of state, are listed in Table 2.

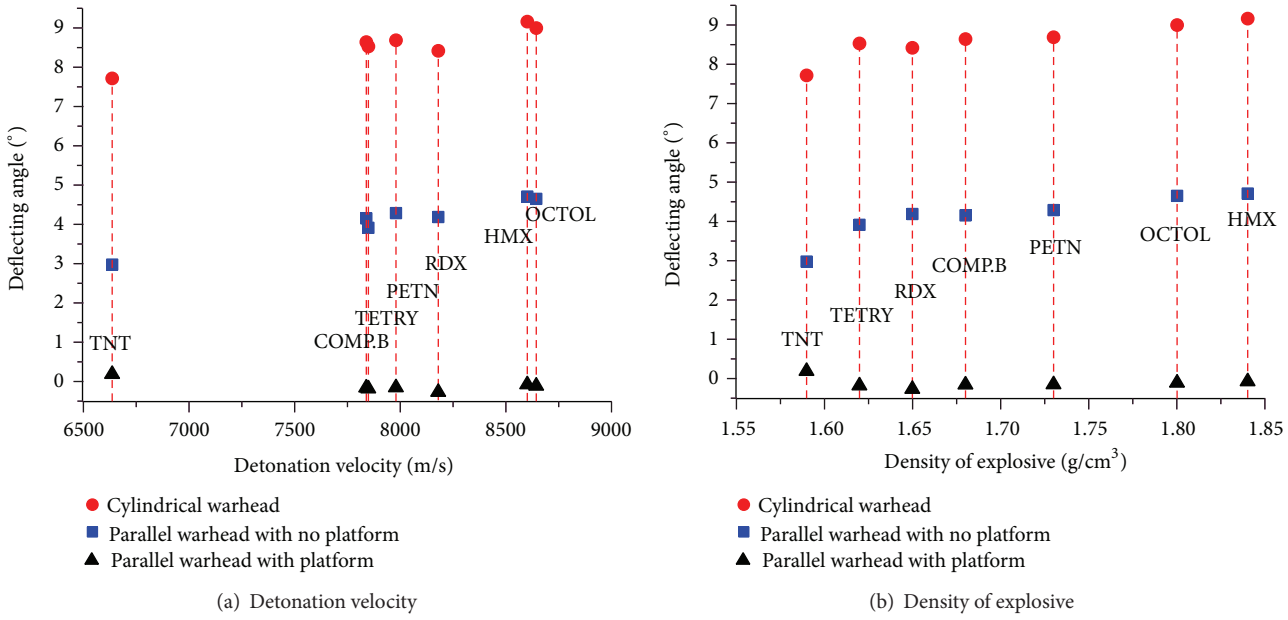
4.2. Factors Influencing Fragment Spatial Distribution. For different kinds of structures, the factors influencing fragment spatial distribution are different and include initiating method, material type, thickness of shells, size of fragment, type of explosive, height of platform, and magnitude of the whole structure. Because the majority of the influencing factors are fixed in this paper, some factors are taken into account for the design of the parallel fragmentation structure.

4.2.1. Type of Explosive. The fragments spraying away from the structure are closely connected to the problems of the explosives loading and responding. The type of explosive and the design of the structure are considerable factors influencing fragment spatial distribution. Seven types of explosives are considered, and the density and the detonation velocity of explosives are shown in Table 1. Three representative kinds of structures are simulated, including the cylindrical structure, the parallel structure with no platform, and the parallel structure with 30 mm height platform. Based on the cylindrical structure with 200 mm height and 200 mm diameter, the other two kinds of structures are modeled.

Figure 14 shows the relationship between the deflecting angle and the density or the detonation velocity of explosives. In Figure 14(a), the deflecting angle of the parallel structure with platform is 0°, approximately. Because of the end effects, the deflecting angle of the structure with no platform increases by 3°~5°, and the angle of the cylindrical structure increases to 9°. Obviously, the design of the platform minimizes the end effects efficiently. Besides, the performance of the explosive does indeed make a difference to the end effects. The higher the energy of explosive is, the stronger the end effects are. As shown in Figure 14, the deflecting angles of the cylindrical structure and the parallel structure with no platform increase in a certain range, increasing the density and the detonation velocity of explosives. However, the type of explosive does not have too much effect on the deflecting angle of the parallel structure with platform. In this paper, we

Table 2: Input data in the numerical simulation.

Material	Part	LS-DYNA material type, material property, and EOS input data (unit = cm, g, μs, Mbar)					
LS-12 aluminum alloy	Cover	*MAT_ELASTIC_PLASTIC_HYDRO_SPALL					
		RO	G	SIGY	EH	PC	FS
		2.73	0.265	$2.95E-3$	$8.384E-3$	−9.0	0.4
1020 steel	Inner shell Outer shell	*MAT_ELASTIC_PLASTIC_HYDRO_SPALL					
		RO	G	SIGY	EH	PC	FS
		7.85	0.618	$7.24E-3$	0.28	−7.03	0.36
Steel	Fragment	*MAT_PLASTIC_KINEMATIC					
		RO	E	PR	SIGY	ETAN	BETA
		7.89	2.069	0.3	$1.724E-2$	0.28	0.36
RDX	Explosive	*MAT_HIGH_EXPLOSIVE_BURN					
		PO	D	PCJ			
		1.65	0.818	0.3			
		*EOS_JWL					
		A	B	$R1$	$R2$	OMEG	$E0$ $V0$
		8.71	0.139	4.6	1.15	0.3	0.1 1.0



(a) Detonation velocity (b) Density of explosive

Figure 14: Deflecting angle of structures with various explosives.

do not focus on the type of explosive, and the explosive of the following structures is RDX.

4.2.2. Height of Platform. The platform is an efficient way to minimize the end effects, but the size of the structure is always limited in a missile. It is necessary to determine the most suitable platform height for the parallel structure designed.

Figure 15 gives the relationship between the height (h) of platform and the deflecting angle (θ) of five structures of varying sizes. It is clear that the influence caused by the platform on the fragment spatial distribution depends on the height of the platform. The end effects become weak with increasing the platform height and can be ignored when the platform height is nearly beyond a value, where the deflecting angle tends to be stable. Referring to the numerical calculation shown in Figure 15, the height of platform is suitable to 30 mm. The 30 mm height platform has minimized the end effects considerably.

4.3. Structure Optimization. The height and diameter of the structures of airborne missiles range from 150 mm to 300 mm. As depicted in Table 3, eighteen parallel fragmentation structures designed by the theoretical method, proposed

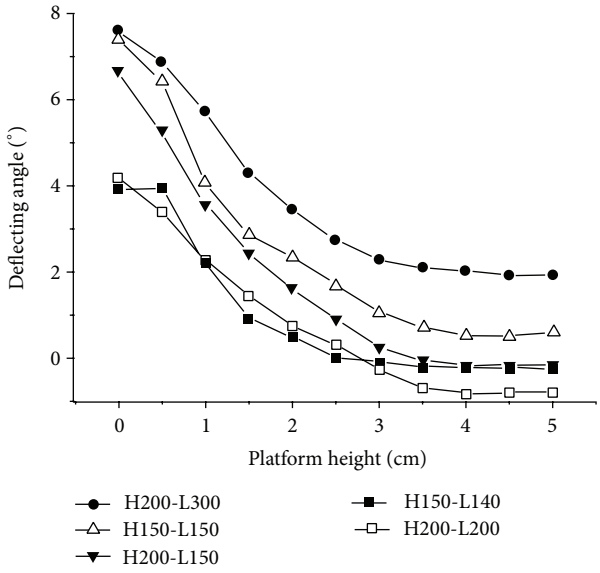


Figure 15: Relationship between the platform height (h) and the deflecting angle (θ).

Table 3: Size of structures and optimum curvature radius.

Number	Height (mm)	Diameter (mm)	Radius (cm)
1	300	300	88.13
2	300	250	97.57
3	300	200	112.52
4	300	150	153.61
5	250	300	76.04
6	250	250	82.76
7	250	200	95.12
8	250	150	128.48
9	200	300	63.00
10	200	250	66.17
11	200	230	68.96
12	200	200	81.15
13	200	150	108.42
14	150	300	48.52
15	150	250	52.08
16	150	200	58.58
17	150	150	74.53
18	150	140	81.71

in this paper, are modeled and the processes that fragments spray away from the shell are simulated.

The design of the shell shape is the key technology of the parallel fragmentation structure. The method of design depicted in Section 2.3 is imperfect, since it ignores the end effects and the structure of the axial hole, so optimizing the curvature radius of the arc equalizing the generatrix of shell is necessary.

The optimum curvature radius differs from the theoretical radius in a range. For each of the eighteen structures,

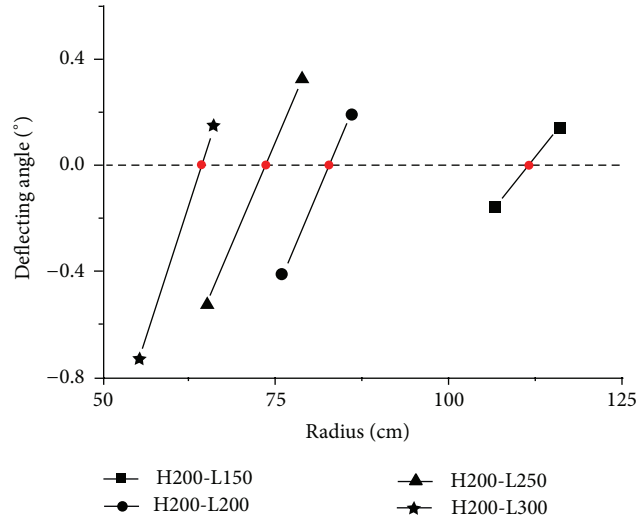


Figure 16: Optimum curvature radius.

Table 4: Parameters of fitted relationship.

z	a	b	c	d	f
72.77	1.01	−0.98	$-6.37E-4$	$2.3E-3$	$-1.63E-3$

we continually rebuild the models which are of different curvature radius around the theoretical radius and simulate the process of spraying fragments. The optimum curvature radius is explored. When the curvature radius of the arc equalizing the generatrix of shell is the optimum curvature radius, the fragment deflecting angle is 0°. The accurate value of the optimum curvature radius is difficult to find, but we can make an approximation. The relationship between the radius and the deflecting angle is depicted in the coordinates shown in Figure 16, with radius as x-axis and deflecting angle as y-axis. A line passes through two points closest to the horizontal line ($Y = 0$) and intersects the line with a third point. The X-coordinate of the intersecting point is the optimum curvature radius. The optimum curvature radii of the eighteen parallel fragmentation structures are depicted in Table 3.

For a parallel fragmentation structure of any size, before the optimized curvature radius is obtained, it is complicated for us to complete a series of modeling and calculation processes. To avoid the complicated process of modeling and calculation, the relationships among the height (H) and diameter (L) of the parallel fragmentation structure and the optimized curvature radius (R) are fitted into an equation as follows:

$$R = z + aH + bL + cH^2 + dL^2 + fHL, \tag{11}$$

where z, a, b, c, d, and f are constant, listed in Table 4. The relationship is fitted by eighteen points depicted in Table 3 and the fitted curved surface is shown in Figure 17. Each point related to the optimized curvature radius can match the curved surface satisfactorily, so (11) can be a reliable reference for the design of the parallel fragmentation structure.

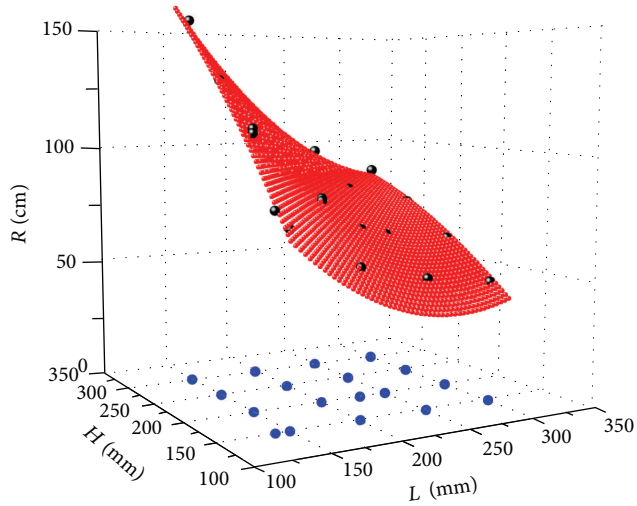


Figure 17: Fitted curved surface.

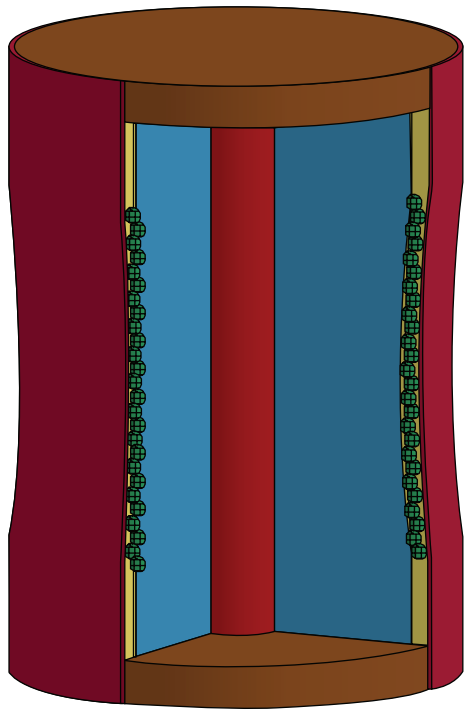


Figure 18: Model of a parallel structure.

4.4. Numerical Verification. In order to verify the veracity and reliability of (11), we design a parallel structure using (11) and simulate the process that fragments spray away from the shell. Based on the numerical information, we can get the fragmentation distributions in the space.

Taking a structure which is in any size into consideration, the height of the structure is 130 mm; the diameter is 157.4 mm. The parameter H and the parameter L in (11) can be expressed as

$$H = 130, \qquad L = 157.4. \tag{12}$$

Taking the above values into (11), we can get that the optimized curvature radius of the structure is 62.68 cm. According to the design above, the parallel structure has 30 mm high platforms on each end, the diameter of fragments is 6 mm, and the thickness of the shells is 1.5 mm. The model of the parallel structure is shown in Figure 18. The structural components and the modeling materials are equal to the model in Section 4.1.

According to the numerical simulation, we can get the process that fragments spray in the space. The side views of fragmentation distributions are shown in Figure 19. It is

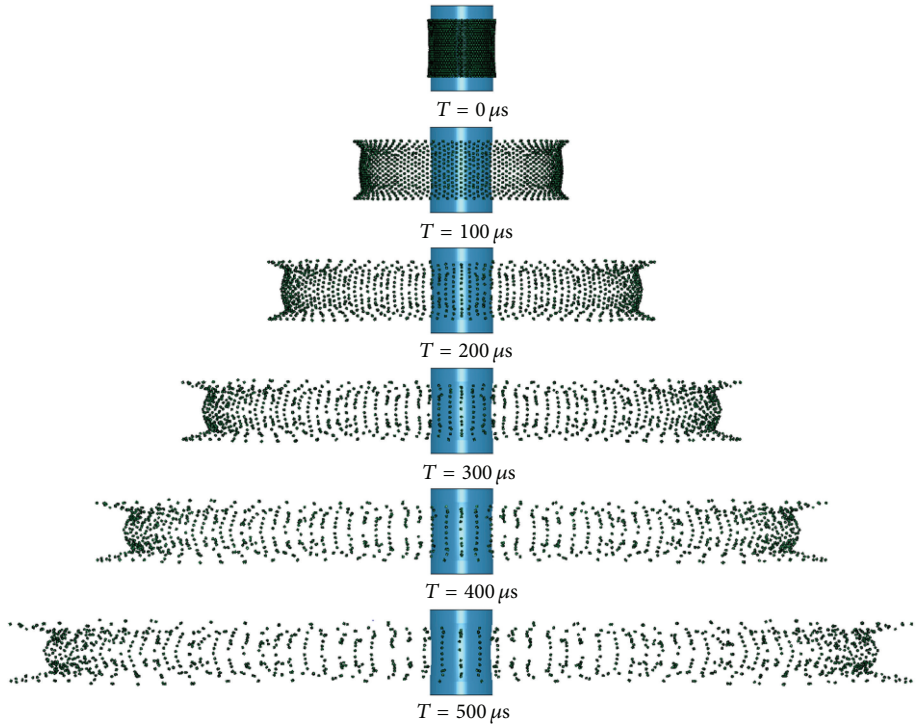


Figure 19: Side view of fragments spraying in the space.

clear that all the fragments keep spraying parallelly. Along axial direction of the parallel structure, the velocities of fragments are zero, and the distributions of fragments range from −6.04 cm to 6.15 cm. The deflecting angle of fragments is −0.04°, which is highly approximate to 0°. It is verified that (11) can be a reliable method for the rapid design of parallel fragmentation structure.

The velocity is an important parameter to evaluate the performance of fragments, as well as the fragmentation distribution. In the model of the parallel fragmentation structure, 26 layer fragments are arranged crisscross in the interlayer between the inner shell and the outer shell. There are 74 fragments in each layer. The number of all the fragments is 1924. At 3.5-meter distance from the initial position of the structure, the average velocity of all the fragments is 1732.19 m/s, and the average velocities of fragments in each layer are different, as shown in Figure 20.

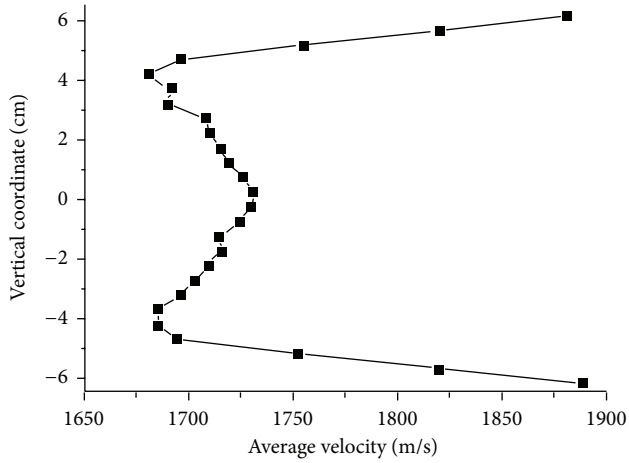


Figure 20: Average velocity of fragments in each layer.

Showing the more features of the parallel fragmentation structure proposed in this paper, the propagation process of detonation wave in the explosive is displayed in Figure 21. In the simulation, the explosive of the model is initiated from the center of the structure. The detonation wave is a spherical wave and reflects on the boundary of explosive at 14 μs.

5. Conclusion

The experiments conducted in this paper in respect of the conventional cylindrical structure adequately indicate that the end effects and detonated method play important roles in the fragment spatial distribution. When the fragment distribution of conventional cylindrical structures, along the axial direction, is in a wide range, the density of fragments on the target is low. For the focusing fragmentation warhead, the ability to focus fragments can only be achieved in a constant distance. Considering the disadvantages of the conventional structure and the specific structure of focusing fragmentation warhead, the parallel fragmentation structure designed in this paper overcomes these disadvantages. The significant characteristic of the parallel fragmentation structure is that fragments spray in parallel, which can not only improve the utilization ratio of fragments, but also achieve cutting damage to the target.

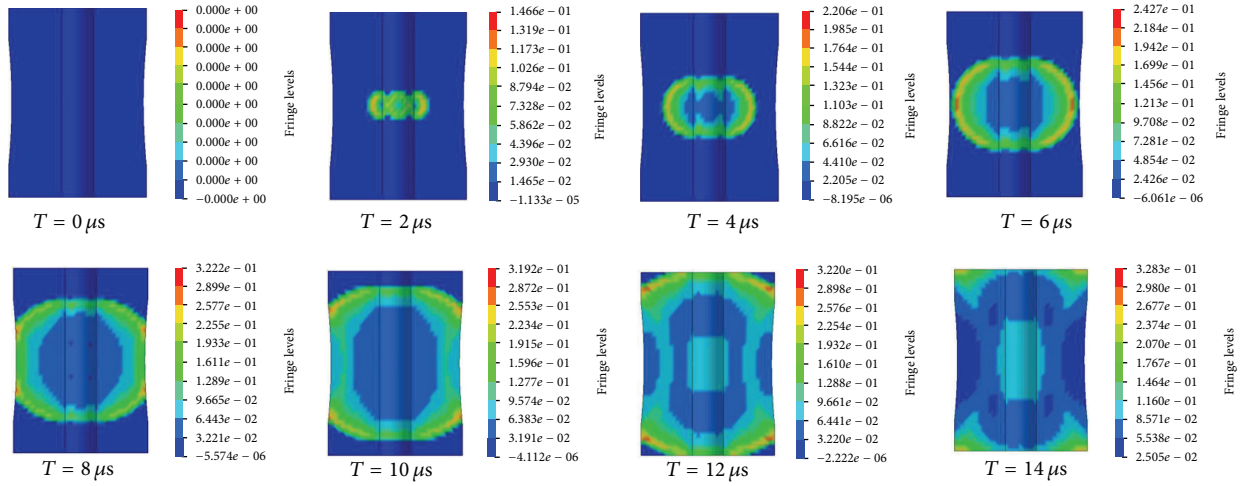


Figure 21: Propagation process of detonation wave in the explosive.

The process that fragments spray away from the shell is studied, and a new method to design the shell shape is used. The theoretical shell shape of the parallel fragmentation structure is calculated through the finite element method. To facilitate the establishment of the numerical model and the machining for relative experiments, the mathematical description for the generatrix of the shell is simplified. There are many factors influencing the distribution of fragments. A number of factors are fixed in this paper and some of the other factors are taken into account.

Regarding the structures of airborne missiles, their height and diameter range from 150 mm to 300 mm. The eighteen parallel fragmentation structures designed through the theoretical method, proposed in this paper, are modeled and the processes that fragments spray away from the shell are simulated. Based on the theoretical design and plentiful simulation data, the relationships between the size of parallel fragmentation structure and the optimized curvature radius of the shell can be represented by an equation. The equation is validated by numerical means and can be a reliable reference to the design of the parallel fragmentation structure.

Conflict of Interests

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

Acknowledgment

This work has been sponsored by the National Natural Science Foundation of China under Grants nos. 11132012 and 11202237.

References

[1] Y. Pei, B.-F. Song, and Q. Han, "Method for assessing the vulnerability of aircraft to spray fragments of missile," *System Engineering Theory and Practice*, vol. 27, no. 2, pp. 161–166, 2007.

[2] J. Li, W. Yang, Y. Zhang, Y. Pei, Y. Ren, and W. Wang, "Aircraft vulnerability modeling and computation methods based on product structure and CATIA," *Chinese Journal of Aeronautics*, vol. 26, no. 2, pp. 334–342, 2013.

[3] M. Held, "Fuze sensor requirements of the different amiable anti air warhead layouts," in *Proceedings of the 23rd International Symposium on Ballistics*, pp. 77–86, Tarragona, Spain, 2007.

[4] J. D. Hughes, "Tactical air warfare of the future," *Journal of Aircraft*, vol. 15, no. 3, pp. 134–136, 1978.

[5] M. Held, "Aimable fragmenting warhead," in *Proceedings of the 13th International Symposium on Ballistics*, pp. 539–548, Stockholm, Sweden, 1992.

[6] G. E. Fairlie, C. O. Marriott, N. J. Robertson, I. H. Livingstone, and N. K. Birnbaum, "Computer modeling of full size fragmenting amiable warheads using AUTODYN-3D," in *Proceedings of the 17th International Symposium on Ballistics*, pp. 389–396, Midrand, South Africa, 1998.

[7] L. Tong, Q. Lixin, and Z. Shouqi, "Study on fragment focusing mode of air-defense missile warhead," *Propellants, Explosives, Pyrotechnics*, vol. 23, no. 5, pp. 240–243, 1998.

[8] G.-Y. Huang, S.-S. Feng, F. Wang, and T. Zhou, "An optical design theory for focused fragmentation warhead," *Defence Science Journal*, vol. 62, no. 4, pp. 205–211, 2012.

[9] S.-S. Feng and G.-Y. Huang, "New design method for focusing fragments of warheads based on theory of optics," in *Proceedings of the 25th International Symposium on Ballistics (ISB '10)*, pp. 926–935, Beijing, China, May 2010.

[10] X.-S. Kong, W.-G. Wu, J. Li, P. Chen, and F. Liu, "Experimental and numerical investigation on a multi-layer protective structure under the synergistic effect of blast and fragment loadings," *International Journal of Impact Engineering*, vol. 65, pp. 146–162, 2014.

[11] J.-Q. Liu, W.-B. Gu, M. Lu, H.-M. Xu, and S.-Z. Wu, "Formation of explosively formed penetrator with fins and its flight characteristics," *Defence Technology*, vol. 10, no. 2, pp. 119–123, 2014.

[12] W. W. Predebon, W. G. Smothers, and C. E. Anderson, "Missile warhead modeling: computations and experiments," Tech. Rep. 2796, Menmorandum, 1977.

[13] F. Lu, B. Jiang, X. Li et al., *The Projection and Damage of Weapon Warhead*, Science Press, Beijing, China, 2013, (Chinese).

[14] P. Weng, G. Ren, and Q. Yu, *Range Experiment of Ammunition*, Weapon Industry Press, Beijing, China, 1996, (Chinese).

[15] R. M. Lloyd, *Physics of Direct Hit and Near Miss Warhead Technologies*, AIAA, Reston, Va, USA, 2001.

[16] R. M. Lloyd, *Conventional Warhead Systems Physics and Engineering Design*, AIAA, Reston, Va, USA, 1998.

[17] S. Sui and S. Wang, *Terminal Effects*, National Defence Industry Press, Beijing, China, 2007, (Chinese).

[18] R. W. Gurney, "The initial velocities of fragments from bombs, shells and grenades," Report 405, Ballistics Research Laboratories, Aberdeen, Md, USA, 1943.

[19] D. K. Crabtree and S. S. Waggener, "Gurney-type formulas for estimating initial fragment velocities for various warhead geometries," Internal Report NSWC TR NO. 86-241, Naval Surface Weapons Center, Dahlgren, Va, USA, 1987.

[20] M. H. Keshavarz and A. Semnani, "The simplest method for calculating energy output and Gurney velocity of explosives," *Journal of Hazardous Materials*, vol. 131, no. 1–3, pp. 1–5, 2006.

[21] A. Zlatkis, N. Korin, and E. Gofman, "Edge effects on fragments dispersion," in *Proceedings of the 23rd International Symposium on Ballistics*, pp. 177–184, Tarragona, Spain, 2007.

[22] P. C. Chou, J. Carleone, W. J. Flis, R. D. Ciccarelli, and E. Hirsch, "Improved formulas for velocity, acceleration and projection angle of explosively driven liners," *Propellants, Explosives, Pyrotechnics*, vol. 8, no. 6, pp. 175–183, 1983.

[23] P. J. Koenig, "A correction for ejection angles of fragments from cylindrical warheads," *Propellants, Explosives, Pyrotechnics*, vol. 12, no. 5, pp. 154–157, 1987.

[24] K. D. Dhote, K. P. S. Murthy, K. M. Rajan, and M. M. Sucheendran, "Quantification of projection angle in fragment generator warhead," *Defence Technology*, vol. 10, no. 2, pp. 177–183, 2014.

[25] G. Randers-Pehrson, "An improved equation for calculating fragment projection angle," in *Proceedings of the 2nd International Symposium on Ballistics*, pp. 223–226, Datona Beach, Fla, USA, 1976.

[26] L. Jing, Z. Wang, and L. Zhao, "Dynamic response of cylindrical sandwich shells with metallic foam cores under blast loading—numerical simulations," *Composite Structures*, vol. 99, pp. 213–223, 2013.

[27] X. Kong, W. Wu, J. Li, F. Liu, P. Chen, and Y. Li, "A numerical investigation on explosive fragmentation of metal casing using Smoothed Particle Hydrodynamic method," *Materials and Design*, vol. 51, pp. 729–741, 2013.

[28] J. Leppänen, "Experiments and numerical analyses of blast and fragment impacts on concrete," *International Journal of Impact Engineering*, vol. 31, no. 7, pp. 843–860, 2005.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 907034, 8 pages
http://dx.doi.org/10.1155/2015/907034

Research Article

An Ant Optimization Model for Unrelated Parallel Machine Scheduling with Energy Consumption and Total Tardiness

Peng Liang,[1,2] Hai-dong Yang,[3] Guo-sheng Liu,[3] and Jian-hua Guo[1]

[1]*School of Computer Science, Guangdong Polytechnic Normal University, Guangzhou 510665, China*
[2]*XingFa Aluminum Holdings Limited, Foshan 528061, China*
[3]*School of Mechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China*

Correspondence should be addressed to Peng Liang; cs_phoenix_liang@163.com

Received 8 October 2014; Accepted 9 January 2015

Academic Editor: Ben T. Nohara

This research considers an unrelated parallel machine scheduling problem with energy consumption and total tardiness. This problem is compounded by two challenges: differences of unrelated parallel machines energy consumption and interaction between job assignments and machine state operations. To begin with, we establish a mathematical model for this problem. Then an ant optimization algorithm based on ATC heuristic rule (ATC-ACO) is presented. Furthermore, optimal parameters of proposed algorithm are defined via Taguchi methods for generating test data. Finally, comparative experiments indicate the proposed ATC-ACO algorithm has better performance on minimizing energy consumption as well as total tardiness and the modified ATC heuristic rule is more effectively on reducing energy consumption.

1. Introduction

In recent years, energy saving has been growing a great interest due to sequence of serious environmental impacts and rising energy cost [1–3]. In manufacturing industry, machine energy consumption can be characterized by power, process time, and state of machines [4]. In particular, a large amount of energy is wasted while keeping idle machine running (i.e., not processing jobs but still running machine) [5–7]. Research on Wichita, an aircraft small-part supplier, shows that at least 13% of total energy consumption can be saved by simply turning off machines while they are not processing any jobs [8]. Kordonowy [9] investigates the background runtime operations of machine and observes that more than 30% of input energy is consumed by background processes. What is more, Drake et al. [10] show that there is a significant amount of energy consumption while machine keeps on idling when no jobs are processed.

As a result, research on minimizing energy consumption with machine operation scheduling should be of benefit to energy saving and reducing carbon dioxide emissions. Only a few references consider the objective of energy consumption [4, 11]. Swaminathan and Chakrabarty [12] considered energy consumption in control systems to extend the life of batteries. Research on Tiwari et al. [13] proved that there is about 40% energy saving when proper power control software is used in microprocessor manufacturing. Mouzon and Yildirim [14] considered the problem of minimizing total energy consumption and total tardiness on signal machine. The total energy consumption is measured by summation of idle power and machine setup power. However, the key to save energy on single machine problem is to determine if the machine should be turned off or not during idle time. Yildirim and Mouzon [15] gave a math mathematical model for minimizing total energy consumption as well as max completion time on signal machine. A conventional genetic algorithm is adopted.

Actually, most of manufacturing systems are unrelated parallel machines. Furthermore, the manager should consider not only the energy consumption costs, but also the due dates of jobs. Ant colony optimization (ACO) algorithm has become more preferable to solve combinatorial optimization problems [16–18]. Yagmahan and Yenisey proposed a multiobjective ant colony system algorithm to solve a flow shop scheduling problem with respect to both of makespan and total flowtime [19]. Lin et al. [20] considered an ACO algorithm to solve the problem of scheduling unrelated

parallel machines to minimize total weighted tardiness. Arnaout et al. [21] addressed the nonpreemptive unrelated parallel machine scheduling problem with machine dependent and sequence-dependent setup times via a modified ACO algorithm. The results showed that ACO outperformed the other algorithms. In this paper, we begin the research of minimizing energy consumption and total tardiness on unrelated parallel machines. The energy consumption of each machine is composed of power cost of machine setup (i.e., machine turning off and then turning on) and power wasted during machine idle period. The problem is formulated by a weighted summation of energy consumption and total tardiness. For solving this problem, we develop an ACO algorithm with ATC rules in which a machine reselection operation is applied.

After this introduction, we describe the problem in Section 2 and the mathematical model is presented in Section 3. The proposed ATC-ACO algorithm is set out in Section 4. Computation results and comparative analysis on 27 test problem configurations and 2187 experiments' results are shown in Section 5. Finally, the main conclusions are included in Section 6.

2. Problem Definition

In this section, a mathematical model is proposed for unrelated parallel machines with the objective of minimizing energy consumption and total tardiness, which is NP-hard, since minimizing energy consumption and total tardiness on single machine is proved to be NP-hard [14]. There are n independent jobs that have to be processed on m parallel machines. Each job can be processed by only one machine and each machine is continuously available. Each job j arrives at time r_j and has a process time p_{ij} on machine i and a due date d_j. The total tardiness is defined as $\sum \max(C_j - d_j, 0)$, where C_j represent the completion time of job j. The machine characteristics are defined as follows. Machine i consumes power P^i_{idle} while machine stands idle. Furthermore, machine i consumes power E^i_{setup} when it is turned off and then turned on (i.e., a setup occurs). To solve this problem, total tardiness and energy consumption must be considered together. If there is a long idle period between two jobs, it may choose to turn off machine to save energy. It means that when the idle energy consumption $P^i_{\text{idle}} * T_{\text{idle}}$ is greater than machine setup energy consumption E^i_{setup}, the machine i will be turned off to save energy. Finally, we conclude the breakeven duration T^i_B is the ratio of machine setup energy consumption E^i_{setup} to machine idle energy consumption P^i_{idle}:

$$T^i_B = \frac{E^i_{\text{setup}}}{P^i_{\text{idle}}}. \tag{1}$$

Unlike single machine scheduling framework proposed by Yildirim and Mouzon [15], unrelated parallel machines scheduling problem is much more complicated. Job assignment is affected not only by the processing time and tardiness cost, but also by the state of machine, which is illustrated on two-machine example in Figure 1. Assume six jobs denoted $\{J_1, J_2, \ldots, J_6\}$ are scheduled on two machines denoted $\{M_1, M_2\}$. The process time p_{ij}, release time r_j, and due date d_j are listed in Table 1. We use horsepower (hp) as the unit of power consumption. The setup energy is defined as $E^1_{\text{setup}} = 5$ hp and $E^2_{\text{setup}} = 9$ hp, idle power consumption is set to $P^1_{\text{idle}} = 1$ hp/sec and $P^2_{\text{idle}} = 2$ hp/sec, and tardiness cost is set to $P^j_{\text{tardiness}} = 1$ hp/sec, $j = 1, 2, \ldots, 6$.

Table 1: Process time, release time, and due date of each job.

Job	J_1	J_2	J_3	J_4	J_5	J_6
p_{1j}	2	5	5	2	8	2
p_{2j}	7	3	9	5	14	2
r_j	12	0	1	10	11	17
d_j	19	7	10	18	22	20

As can be seen in Figure 1, a feasible solution is decided by making three decisions: machine assignment, job sequencing, and machine state (idling or from turning off to turning on). According to the definition of setup energy, tardiness, and idle power consumption, the breakeven durations T^1_B, T^2_B are 5 sec and 4.5 sec, respectively. Whether keeping machine idle or performing a machine setup depends on the comprising breakeven duration T^i_B with waiting times between jobs. Furthermore, assigning jobs on machines relies not only on processing cost and machine available time, but also on setup energy and idle power consumption. Note that J_6 arrived at time 17, the tardiness of J_6 is 2 hp in solution 1, while idle power consumption between J_1 and J_6 is 3 hp in solution 2. In order to select an appropriate solution minimization of energy consumption and total tardiness on unrelated parallel machines, an ant colony optimization framework is proposed.

3. Mathematical Model

Basic Notions

m: the number of machines;

n: the number of jobs;

J_j: the job j, $j = 1, 2, \ldots, n$;

M_i: the machine i, $i = 1, 2, \ldots, m$;

H_i: the number of jobs allocated on machine M_i;

w_1: weight associated with total tardiness;

w_2: weight associated with energy consumption;

c_j: the completion time of job J_j;

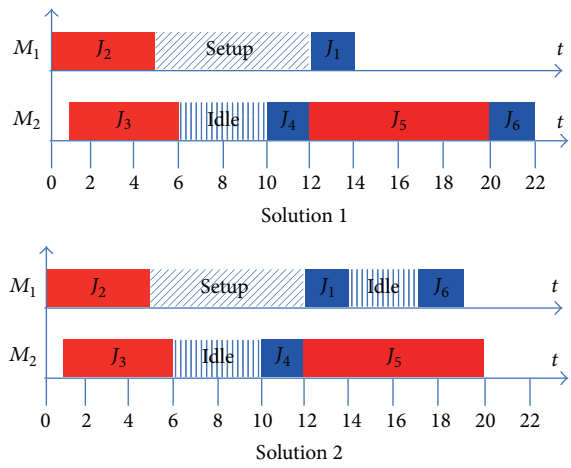
r_j: the release time of job J_j;

d_j: the due date of job J_j;

t_i: the makespan of scheduled jobs on machine M_i;

p_{ij}: the process time of job J_j on machine M_i;

$P^j_{\text{tardiness}}$: per unit time cost of job J_j tardiness;



Solution	Total energy consumption	Total tardiness
Solution 1	13 hp	2 hp
Solution 2	16 hp	0

FIGURE 1: An illustration of a feasible solution for scheduling on two unrelated parallel machines with six jobs.

P^i_{idle}: per unit time energy consumption of machine M_i.

Decision Variables. Consider

$$y_{ilk} = \begin{cases} P^i_{\text{idle}}\left(c_k - p_{ik} - c_l\right), & \text{If } \left(c_k - p_{ik} - c_l\right) < T^i_B \\ & \text{and job } J_l \text{ precedes job } J_k \\ E^i_{\text{setup}}, & \text{otherwise.} \end{cases} \tag{2}$$

The definition of minimizing energy consumption and total tardiness on unrelated parallel machines is formulated as follows:

$$\text{Min}\left(w_1 \sum_{j=1}^{n} P^j_{\text{tardiness}} \max\left(0, c_j - d_j\right) + w_2 \sum_{i=1}^{m} \sum_{k=1,k\neq j}^{H_i} \sum_{l=1}^{H_i} y_{ilk}\right) \tag{3}$$

$$c_j - p_{ij} \geq r_j \tag{4}$$

$$c_l - p_{il} \geq c_k \quad \text{or} \quad c_k - p_{ik} \geq c_l \tag{5}$$

$$\text{If } \left(c_k - p_{ik} - c_l\right) < T^i_B \text{ and job } J_l \text{ immediately precedes job } J_k \text{ on machine } i \text{ then} \tag{6}$$

$$y_{ilk} = P^i_{\text{idle}}\left(c_k - p_{ik} - c_l\right), \quad \text{else } y_{ilk} = E^i_{\text{setup}}$$

$$\forall i = 1, 2, \ldots, m \quad \forall j = 1, 2, \ldots, n$$

$$\forall l = 1, 2, \ldots, H_i \quad \forall k = 1, 2, \ldots, H_i, \quad k \neq l. \tag{7}$$

Our multiobjective function is started in (3) which aims at minimizing the weighted summation of energy consumption and total tardiness. Constraint (4) guarantees that a job cannot be processed before it is released. Constraint (5) ensures that only one job could be processed on each machine at the same time. Constraint (6) defines that if waiting time between job J_l and job J_k (job J_l precedes job J_k) on machine M_i is longer than machine breakeven duration T^i_B, then y_{ilk} is equal to the corresponding machine setup energy consumption or otherwise equal to the corresponding machine idle power consumption.

4. Ant Colony Optimization Algorithm Based on ATC Heuristic Rule (ACO-ATC)

The ACO algorithm imitates the indirect communications within artificial ants to find the shortest path between food and their net. These communications are recorded by artificial pheromone trails. Naturally, pheromone in long paths will evaporate much quicker than short paths, and then short paths will attract more ants for denser pheromone. In this section, we propose an ACO-ATC algorithm to solve the problem of scheduling unrelated parallel machines to minimize energy consumption and total tardiness. Details of proposed algorithm are described in the following subsections.

4.1. Solution Construction. The solution component for scheduling unrelated parallel machines to minimize energy consumption and total tardiness required two decisions: assignment and job sequence, which will result in a huge solution space. Consequently, the two decisions are often addressed independently to reduce the solution search space, such as selecting the first available machine and then distributing the minimization total tardiness job. Finally, after the solution is constructed, machine states are fixed according to the job sequence.

Although this strategy could significantly reduce the search space, appealing solutions may be excluded due to the independent decision heuristic. As can be seen in Figure 1, the available machine (M_1) selected in the first scheduled strategy may not be the minimization energy consumption and total tardiness for selected job (J_6) in the second scheduled strategy. Inspired by ATC heuristic rule proposed by Lin et al. [20], we modify a new solution construction mechanism requiring three step: first, machine selection,

then job selection, and finally machine reselection. Details of modified solution construction are shown as follows.

4.1.1. Machine Selection. First, a machine will be selected. We generate a random number q_m from uniform distribution [0, 1]. A user-specified number $q_{m0} = 0.9$ represents the relative importance of exploitation versus exploration. If $q_m < q_{m0}$ an ant is apt to select the smallest makespan machine among all unrelated parallel machines according to (8); otherwise a machine I is chosen according to the probability distribution P_i defined in (9)

$$i^* = \begin{cases} \min\limits_{1\le i\le m} t_i, & \text{if } q_m < q_{m0} \\ I, & \text{otherwise,} \end{cases} \tag{8}$$

$$P_i = \frac{1/t_i}{\sum_{p=1}^{m} 1/t_p} \quad i = 1, 2, \ldots, m. \tag{9}$$

4.1.2. Job Selection. A job will be selected after a machine has been chosen. Job selection defined in (10) considers the heuristic information and pheromone trails together. We generate a random number q_j from uniform distribution [0, 1]. $q_{j0} = 0.9$ is a user-specified number. If $q_j < q_{j0}$ an ant is apt to select the smallest tardiness job j processed on machine i^* according to (10); otherwise a job J is chosen according to the probability distribution P_{i^*j} defined in (11). Pheromone trails $\tau_{i^*j}(t)$ indicate the favorability of assigning job j to a machine i^* and set to 0 initially. $\eta_{i^*j}(t)$ is heuristic information which suggests the greedy heuristic of processing the job j on machine i^* that takes the least amount of tardiness, which is presented in (12). Parameters α and β are the relative importance of pheromone trails and heuristic information, respectively. Ψ represents a set of unscheduled jobs in (11)

$$j^* = \begin{cases} \max\limits_{j} \left(\left[\tau_{i^*j}(t)\right]^{\alpha} \cdot \left[\eta_{i^*j}(t)\right]^{\beta} \right), & \text{if } q_j < q_{j0} \\ J, & \text{otherwise,} \end{cases} \tag{10}$$

$$P_{i^*j} = \begin{cases} \dfrac{\left[\tau_{i^*j}(t)\right]^{\alpha} \cdot \left[\eta_{i^*j}(t)\right]^{\beta}}{\sum_{l\in\Psi} \left[\tau_{i^*l}(t)\right]^{\alpha} \cdot \left[\eta_{i^*l}(t)\right]^{\beta}}, & \text{if } j \in \Psi \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

$$\eta_{i^*j}(t) = \frac{1}{P^j_{\text{tardiness}} \times \max\left\{t_{i^*} + p_{ij^*} - d_{j^*}, 0\right\} + 1}. \tag{12}$$

4.1.3. Machine Reselection. Since the computation of energy consumption and total tardiness need to confirm machine and job sequence simultaneously, the independent selection strategy may not find the appealing solution. In order to solve this problem, a machine reselection will be executed after job j^* has been selected. An ant will select machine i^{**} according to (13), which aims at minimizing the weighted sum of energy consumption and total tardiness when processing job j^* on machine i^{**}:

$$\begin{aligned} i^{**} = \arg\min \Big\{ & U_{ij^*} \times \min\left\{E^i_{\text{setup}}, P^i_{\text{idle}}\left(r_{j^*} - t_i\right)\right\} \\ & + P^j_{\text{tardiness}}\left(1 - U_{ij^*}\right) \\ & \times \max\left\{t_i + p_{ij^*} - d_{j^*}, 0\right\}\Big\} \end{aligned} \tag{13}$$

$$U_{ij^*} = \begin{cases} 0, & \text{if } r_{j^*} \le t_i \\ 1, & \text{otherwise.} \end{cases}$$

After three steps (machine selection, job selection, and machine reselection) are executed, a job j is assigned to a machine i^*. Repeat the operations above until all jobs are distributed; then a solution construction is finished.

4.2. Local Search. Dorigo and Stützle [22] have proved that ACO algorithm may be further improved by incorporating an appropriate local search algorithm. Therefore, we include two local search strategies (LS1 and LS2) in our implementation of ACO-ATC algorithm.

The first procedure (LS1) searches for new solutions by swapping jobs on the same machine. The second procedure (LS2) searches for new solution by transferring jobs from the machine with the highest objective value to the machine with the lowest one. The computation of its implementation is $O(m * n^2)$. The pseudocode for local search algorithm is summarized in Pseudocode 1.

4.3. Pheromone Update. Once all ants have constructed their solutions, global pheromone updating rules are performed. Initially, there are no pheromone trails on all solutions. The global pheromone updating rules are defined as follows:

$$\tau_{ij} = (1 - \rho)\,\tau_{ij} + \rho\Delta\tau_{ij},$$

$$\Delta\tau_{ij} = \begin{cases} \dfrac{1}{L_{\text{best}}}, & \text{if } (i, j) \in \text{best solution} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Global updating is intended to provide more pheromone to the best performance solution. Pheromone evaporation rate ρ ($0 < \rho < 1$) is used to forget bad solutions and to explore new solutions. The pheromone amount of all solution components is updated by increasing the reciprocal of the best objective value L_{best}.

5. Computational Experiments and Results

5.1. Data Generation. In this section, the data of computational experiments will be presented to evaluate the proposed ACO-ATC algorithm. The proposed algorithm is implemented in Matlab R2012b running on Windows 7 with Intel core i5 2.30 GHz and 4 Gigabytes RAM. The number of jobs and number of machines are divided into three different sizes, namely, small, medium, and large, which take the value of 20/5, 50/5, and 50/10, respectively. Processing times p_{ij} are generated randomly from uniform distribution

```
Local Search Algorithm:
Set IterationNum = 1;
While (IterationNum < MaxIterationNum)
  For each machine i
    For each job j1 in machine i
      For each job j2 (j2 ≠ j1) in machine i
        Construct new solution by exchanging two jobs
        If the new solution is better than current one, then exchange two jobs
  Find the machine i1 with the highest objective value
  Find the machine i2 with the lowest objective value
  For each j1 in machine i1
    For each j2 in machine i2
      Construct new solution by exchanging two jobs
      If the new solution is better than current one, then exchange two jobs
```

Pseudocode 1

Table 2: Parameter setting of the main factors in experimental design.

Factor	Count	Levels		
Job n/machine m	3	20/5	50/5	100/10
c	3	2	4	8
$E^i_{\text{setup}}/P^i_{\text{idle}}$	3	2	4	8

[1, 3]. Release times r_j are generated randomly from uniform distribution [1, 30]. Due dates of jobs d_j are generated by TWK (total work-content) method and calculated by (15), where c represents the relaxation coefficient and is set to 2, 4, and 8. As c increases, the difference between due dates and release times becomes larger, which means that the problem becomes less constrained and easily solved. The per unit time of job tardiness cost $P^j_{\text{tardiness}}$ is calculated by (16). We generate the unit time of machine idle power consumption P^i_{idle} randomly from uniform distribution [1, 3]. The state (idle or from turning off to turning on) of machine only depends on the setting of unit time of machine idle energy consumption P^i_{idle} and machine setup energy consumption E^i_{setup}. We use $E^i_{\text{setup}}/P^i_{\text{idle}}$ ratio to define this instance and set $E^i_{\text{setup}}/P^i_{\text{idle}}$ ratios to 2, 4, and 8. Equal relative weightings chosen for total tardiness and energy consumption for total objective value are $w_1 = w_2 = 0.5$, respectively. All the parameter settings of each main factor are shown in Table 2. Consider

$$d_j = r_j + c \times \sum_{i=1}^{m} \frac{p_{ij}}{m}, \tag{15}$$

$$P^j_{\text{tardiness}} = \sum_{i=1}^{m} \frac{p_{ij}}{m}. \tag{16}$$

When the data are generated, all the level combinations result in $3 * 3 * 3 = 27$ test problem configurations.

In order to evaluate the performance of proposed ACO-ATC algorithm, we first compare the ACO-ATC algorithm with a comparative algorithm named GRASPTETT, which is a multiobjective algorithm to solve the minimization problem of energy consumption and total tardiness on single machine. For more details of GRASPTETT, see Mouzon and Yildirim [14]. We extend the GRASPTETT algorithm on unrelated parallel machines in this paper by using a well-known earliest release time heuristic to assign machines. What is more, to validate our modified ATC heuristic rule (machine selection, job selection, and machine reselection), we also compare ACO-ATC with original ACO algorithm (OACO) which only adopts "machine first, schedule job second" solution construction strategy. We incorporate the same parameter setting of OACO and other important parameters ($E^i_{\text{setup}}/P^i_{\text{idle}}$ and c) in this paper.

5.2. Performance Measure. The relative percentage deviation (RPD) is used to evaluate the performance of multiobjective optimization algorithms. Given an obtained objective value by selected optimization algorithm, the RPD can be defined in (17) as follows:

$$\text{RPD} = \frac{\text{Value}_{\text{sel}} - \text{Min}_{\text{sol}}}{\text{Min}_{\text{sol}}} \times 100\%, \tag{17}$$

where Min_{sol} is the best objective value obtained for each problem configuration.

5.3. Parameter Tuning. Since the parameters of ACO algorithm significantly influence computation results, Taguchi method [23, 24] is utilized to determine the appropriate values for ACO parameters that minimize the objective value for each problem configuration. The factors considered in parameter tuning experiment are as follows: β (0.01, 0.15, 0.3), *Num_Ants* (5, 20, 40), ρ (0.01, 0.15, 0.3), and α (0.01, 0.15, 0.3). To reduce the number of runs but reach sound conclusions, the orthogonal array L_9 described in Taguchi method is chosen according to the number of parameters and the number of factor levels. For each problem configuration, three instances are generated where each instance is run 3 times independently for each parameter combination, which means that we have to do $27 * 3 * 3 * 9 = 2187$ experiments, and the average objective value (AOV) is obtained for each

Table 3: Orthogonal array and AOV results.

Experiment number	Factor				AOV
	β	*Num_Ants*	ρ	α	
1	1	1	1	1	198.9
2	1	2	2	2	204.7
3	1	3	3	3	210.3
4	2	1	2	3	198.3
5	**2**	**2**	**3**	**1**	**194.7**
6	2	3	1	2	197.6
7	3	1	3	2	204.2
8	3	2	1	3	213.6
9	3	3	2	1	208.4

Table 4: Response value and significance rank of each parameter.

Level	β	*Num_Ants*	ρ	α
1	204.6	200.4	203.4	200.7
2	196.9	204.3	208.3	202.2
3	208.7	205.4	203.1	207.4
Delta	11.8	5	5.2	6.7
Rank	1	4	3	2

problem configuration. We implement the Taguchi method by using the small size configuration where $n = 20$, $m = 5$, $c = 4$, and $E^i_{\text{setup}}/P^i_{\text{idle}} = 4$. The orthogonal array and AOV results are listed in Table 3, where the second column of Table 3 represents $\beta = 0.01$, *Num_Ants* $= 20$, $\rho = 0.15$, and $\alpha = 0.15$.

According to the orthogonal array and AOV results, we can analyze the importance of each factor with its response value and significance rank, which is shown in Table 4. As can be seen in Table 4, heuristic information parameter β is the most significant one among all parameters. It means that heuristic information for machine selection and job sequence is crucial to the proposed ACO-ATC algorithm. An appropriate value of β could lead to better convergence stability. Since the parameter α ranks second, it implies that the amount of pheromone amplification is also important. A small value of ρ will lead to a faster convergence rate and a small value *Num_Ants* is enough for searching the solution space. According to the analysis above, for problem configuration, $n = 50$, $m = 10$, $c = 4$, and $E^i_{\text{setup}}/P^i_{\text{idle}} = 4$, a good choice of parameter combination is suggested as $\beta = 0.15$, *Num_Ants* $= 5$, $\rho = 0.01$, and $\alpha = 0.01$.

5.4. Comparative Results. In this section, our proposed ACO-ATC algorithm is tested on all 27 problem configurations. Each problem configuration generates 3 instances and each test is repeated with 5 runs for each instance. Parameter settings are the same as discussed in the last section (see Section 5.3). Performance of solutions to yield using test problem is compared with two multiobjective optimization algorithms: GRASPTETT and OACO. The computational results of average RPD for all problem configurations are shown in Table 5, respectively.

As can be seen in Table 5, the ACO-ATC algorithm performs better than the other two approaches in all problem configurations. The mean RPD values of all three algorithms are consistent when job number and machine number n/m are increasing. The mean RPD value for all tests of ACO-ATC algorithm is 0.96, when the mean RPD value of GRASPTETT algorithm is 4.85 which is 3.89 higher than ACO-ATC algorithm. The OACO algorithm shows the weakest performance with 5.95 mean RPD. The factor n/m has significant influence on GRASPTETT and OACO. Furthermore, machine reselection heuristic rule is crucial for solution construction since ACO-ATC algorithm outperforms OACO in all instances. When relaxation coefficient c is small ($c = 2$), the performances of GRASPTETT and OACO are acceptable, especially in small problem size where $n = 20$ and $m = 5$, for the reason that there is only little scheduling space when the due dates are not well spread and waiting time between release time and due date is small. With the increasing of relaxation coefficient c, the differences between ACO-ATC and compared approaches become larger, for the reason that the bigger the margin between release time and due date, the less the probability of job tardiness occurrence. By increasing the ratio of $E^i_{\text{setup}}/P^i_{\text{idle}}$, which means to increase the length of breakeven duration, all approaches have a little fluctuation. It could be explained by machines trend to remain idle in short waiting time.

6. Conclusion

In this study we have successfully implemented the problem of minimizing energy consumption and total tardiness on unrelated parallel machines. Due dates and release times are distinct, and the breakeven duration of each machine is different. A compromised balance has to be found between machine energy consumption and total tardiness. We proposed a framework with an ant colony optimization algorithm (ACO) and ATC heuristic rule to solve this problem. Furthermore, it is a new kind of problem for minimization of machine energy consumption and total tardiness on unrelated parallel machines which need to be modeled and solved effectively.

In the computation evaluation, two approaches (GRASPTETT and OACO) for solving minimizing machine energy consumption and total tardiness on single machine are adapted and compared with proposed ACO-ATC algorithm. The ATC-ACO algorithm outperforms other approaches and GRASPTETT shows better than OACO in most of instances.

Although this work has dealt with several challenging issues, future work is still needed. Firstly, more machine states should be considered (e.g., machine has a warm-up time which depends on the length of setup time). In this situation, the breakeven duration is variable according to the setup time, which will make problem much more complicated. The second extension should obtain an approximate Pareto front via Pareto ACO algorithm, and then the decision maker can select a suitable choice among all solutions.

Table 5: Comparative results of three multiobjective optimization algorithms.

n/m	c	$E^i_{\text{setup}}/P^i_{\text{idle}}$	Proposed ACO (average RPD)	GRASPTETT (average RPD)	OACO (average RPD)
20/5	2	2	0.46	0.62	2.12
		4	0.58	1.50	2.66
		8	0.31	2.35	4.12
	4	2	1.27	2.45	3.98
		4	0.41	3.57	5.60
		8	0.52	5.78	6.76
	8	2	1.36	3.78	4.39
		4	0.67	3.30	6.63
		8	0.33	6.08	7.47
Mean			0.66	3.27	4.86
50/5	2	2	1.15	2.60	2.88
		4	0.97	3.13	3.28
		8	0.35	4.35	4.24
	4	2	1.64	4.59	5.82
		4	0.64	5.77	6.59
		8	0.28	6.96	8.13
	8	2	1.85	4.84	3.45
		4	1.07	5.42	7.54
		8	0.44	6.20	8.80
Mean			0.93	4.87	5.64
100/10	2	2	2.08	4.75	3.76
		4	1.36	6.35	6.64
		8	0.48	7.23	7.30
	4	2	1.45	3.92	5.24
		4	1.09	6.85	7.49
		8	0.52	7.20	9.10
	8	2	2.59	4.77	6.09
		4	1.71	7.48	9.46
		8	0.34	9.24	11.08
Mean			1.29	6.42	7.35
Mean for all			0.96	4.85	5.95

Conflict of Interests

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

Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 51475096) and Guangdong Province Key Scientific and Technological Project, no. 2012A080104022.

References

[1] Energy Information Administration, *Annual Energy Review*, 2004, http://www.eia.doe.gov/emeu/aer/consump.html.

[2] Solar Energy International, Energy Facts, 2006.

[3] BMWi, "Energy Statistics," German Federal Ministry of Economics and Technology, 2009.

[4] G. Mouzon, M. B. Yildirim, and J. Twomey, "Operational methods for minimization of energy consumption of manufacturing equipment," *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4247–4271, 2007.

[5] J. B. Dahmus and T. C. Gutowski, "An environmental analysis of machining," in *Proceedings of the ASME International Mechanical Engineering Congress and RD&D Expo (IMECE '04)*, pp. 1–10, Anaheim, Calif, USA, November 2004.

[6] T. Simunic, L. Benini, P. Glynn, and G. D. Micheli, "Dynamic power management for portable systems," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pp. 11–19, Boston, Mass, USA, August 2000.

[7] T. Gutowski, C. Murphy, D. Allen et al., "Environmentally benign manufacturing: observations from Japan, Europe and

the United States," *Journal of Cleaner Production*, vol. 13, no. 1, pp. 1–17, 2005.

[8] R. Drake, M. B. Yildirim, J. Twomey, L. Whitman, J. Ahmad, and P. Lodhia, "Data collection framework on energy consumption in manufacturing," in *Proceedings of the Industrial Engineering Research Conference*, pp. 1–7, Orlando, Fla, USA, May 2006.

[9] D. N. Kordonowy, *A power assessment of machining tools [B.S. thesis]*, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2002.

[10] R. Drake, M. B. Yildirim, J. Twomey, L. Whitman, J. Ahmad, and P. Lodhia, "Data collection framework on energy consumption in manufacturing," in *Proceedings of the Industrial Engineering Research Conference*, Orlando, Fla, USA, May 2006.

[11] H. Hoogeveen, "Multicriteria scheduling," *European Journal of Operational Research*, vol. 167, no. 3, pp. 592–623, 2005.

[12] V. Swaminathan and K. Chakrabarty, "Energy-conscious, deterministic I/O device scheduling in hard real-time systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 7, pp. 847–858, 2003.

[13] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, no. 4, pp. 437–445, 1994.

[14] G. Mouzon and M. B. Yildirim, "A framework to minimise total energy consumption and total tardiness on a single machine," *International Journal of Sustainable Engineering*, vol. 1, no. 2, pp. 105–116, 2008.

[15] M. B. Yildirim and G. Mouzon, "Single-machine sustainable production planning to minimize total energy consumption and total completion time using a multiple objective genetic algorithm," *IEEE Transactions on Engineering Management*, vol. 59, no. 4, pp. 585–597, 2012.

[16] C.-Y. Low, R.-K. Li, and G.-H. Wu, "Ant colony optimization algorithms for unrelated parallel machine scheduling with controllable processing times and eligibility constraints," in *Proceedings of the Institute of Industrial Engineers Asian Conference*, pp. 79–87, Taipei, Taiwan, 2013.

[17] M. Randall and A. Lewis, "A parallel implementation of ant colony optimization," *Journal of Parallel and Distributed Computing*, vol. 62, no. 9, pp. 1421–1432, 2002.

[18] T. Stützle, "Parallelization strategies for ant colony optimization," in *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN '98), Amsterdam, The Netherlands, September 1998*, A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., vol. 1498 of *Lecture Notes in Computer Science*, pp. 722–741, Springer, Berlin, Germany, 1998.

[19] B. Yagmahan and M. M. Yenisey, "A multi-objective ant colony system algorithm for flow shop scheduling problem," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1361–1368, 2010.

[20] C.-W. Lin, Y.-K. Lin, and H.-T. Hsieh, "Ant colony optimization for unrelated parallel machine scheduling," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1–4, pp. 35–45, 2013.

[21] J.-P. Arnaout, G. Rabadi, and R. Musa, "A two-stage Ant Colony Optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times," *Journal of Intelligent Manufacturing*, vol. 21, no. 6, pp. 693–701, 2010.

[22] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2004.

[23] R. K. Roy, *A Primer on the Taguchi Method*, Society of Manufacturing Engineers Press, New York, NY, USA, 2nd edition, 2010.

[24] R. S. Rao, C. G. Kumar, R. S. Prakasham, and P. J. Hobbs, "The taguchi methodology as a statistical tool for biotechnological applications: a critical appraisal," *Biotechnology Journal*, vol. 3, no. 4, pp. 510–523, 2008.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 240419, 17 pages
http://dx.doi.org/10.1155/2015/240419

Research Article

An Effective Hybrid of Bees Algorithm and Differential Evolution Algorithm in Data Clustering

Mohammad Babrdel Bonab,[1] Siti Zaiton Mohd Hashim,[1] Nor Erne Nazira Bazin,[1] and Ahmed Khalaf Zager Alsaedi[2]

[1]*Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia*
[2]*College of Science, Misan University, Ministry of Higher Education of Iraq, Iraq*

Correspondence should be addressed to Mohammad Babrdel Bonab; bbmohammad2@live.utm.my

Received 8 October 2014; Accepted 16 January 2015

Academic Editor: Yi-Chung Hu

Clustering is one of the most commonly used approaches in data mining and data analysis. One clustering technique in clustering that gains big attention in clustering related research is k-means clustering such that the observation is grouped into k cluster. However, some obstacles such as the adherence of results to the initial cluster centers or the risk of getting trapped into local optimality hinder the overall clustering performance. The purpose of this research is to minimize the dissimilarity of all points of a cluster from gravity center of the cluster with respect to capacity constraints in each cluster, such that each element is allocated to only one cluster. This paper proposes an effective combination algorithm to find optimal cluster center for the analysis of data in data mining and a new combination algorithm is proposed to untangle the clustering problem. This paper presents a new hybrid algorithm, which is, based on cluster center initialization algorithm (CCIA), bees algorithm (BA), and differential evolution (DE), known as CCIA-BADE-K, aiming at finding the best cluster center. The proposed algorithm performance is evaluated with standard data set. The evaluation results of the proposed algorithm and its comparison with other alternative algorithms in the literature confirm its superior performance and higher efficiency.

1. Introduction

Data clustering is one of the most important knowledge discovery techniques to extract structures from dataset and is widely used in data mining, machine learning, statistical data analysis, vector quantization, and pattern recognition. The aim of clustering is to partition data into k cluster, so that each cluster contains data, which has the most similarity and maximum dissimilarity with the other clusters. Clustering algorithms can be comprehensively classified into hierarchical, partitioning, model-based, grid-based, and concentration-based clustering algorithms [1–3].

Hierarchical clustering algorithm divides a dataset into a number of levels of nested partitioning. In the partitioning algorithms observations of one dataset decompose into a set of k clusters with most similarity among intra-group members and least similarity among inter group members [4]. Dissimilarities are evaluated based on attribute values. Generally, distance criterion is used for data analysis [5].

The k-means algorithm is one of the partitional clustering algorithm and one of the most popular algorithms, used in many domains. The k-means algorithm implementation is easy and often practical. However, results of k-means algorithm considerably depend on initial state. In other words, its efficiency highly depends on the first initial center [6].

The main purpose of k-means clustering algorithm is to minimize the diversity of all objects in a cluster from their cluster centers. The initialization problem of k-means algorithm is considered by heuristic algorithms, but it still risks being trapped in local optimality. Therefore, for achieving a better cluster algorithm we should find a solution for overcoming the problem of trap into local optimum [7].

There are many studies to overcome this problem. For instance, Niknam and Amiri have proposed a hybrid approach based on combining partial swarm optimization and ant colony optimization with k-means algorithm for data clustering [8], and Nguyen and Cios have proposed a combination technique based on the hybrid of k-means, genetic algorithm, and maximization of logarithmic regression expectation [9]. Kao et al. have presented a combination algorithm according to the hybrid of partial swarm optimization, Nelder-Mead simplex search and genetic algorithm [10]. Krishna and Murty proposed an algorithm for cluster analysis called genetic k-means algorithm [11]. Žalik proposed an approach for clustering without preassigning cluster numbers [12]. Maulik and Bandyopadhyay haves introduced genetic based algorithm to solve this problem and evaluate the performance on real data. They define spatial distance-based mutation according to mutation operator for clustering [13]. Laszlo and Mukherjee have proposed another genetic based approach, that for k-means clustering exchanges neighboring cluster centers [14]. Fathian et al. have presented a technique to overcome clustering problem according to honey-bees mating optimization (HBMO) [15–17]. Shelokar et al. have presented to solve clustering problem based on the ant colony optimization [18]. Niknam et al., have combined to dominate this problem based on the simulated annealing and ant colony optimization [19]. Ng and Sung have introduced a technique based on the taboo search to find cluster center [20, 21]. Niknam et al. have introduced a hybrid approach based on combining partial swarm optimization and ant simulated annealing to solve clustering problem [22, 23].

The bees algorithms can be classified in two main categories including foraging-based honeybee algorithms and marriage-based honeybee algorithm. Each of these categories have many algorithm such as artificial bee algorithm (ABC) [3, 24, 25], corporate artificial bee algorithm (CABC) [26], parallel artificial bee algorithm (PABC) [27], bee colony optimization (BCO) [28, 29], bee algorithm (BA) [30], bee foraging algorithm (BFA) [31], bee swarm optimization (BSO) for first categories [32]. Marriage in honey-bees optimization (MBO) [32], fast marriage honey-bees optimization (FMBO) [33], and finally modified fast marriage in honey-bees optimization (MFMBO) are in the second category of bee algorithm [34].

One of the foraging-based algorithms is the bees algorithm that is a new population based search algorithm, developed by Pham et al. in 2006 [30]. The algorithm mimics the food foraging behavior of swarms of honeybees (Figure 3). In its basic version, the algorithm performs a kind of neighborhood search combined with random search and can be used for optimization problems [30].

Differential evolution is an evolutionary algorithm (EA), which has been widely used in to optimization problems, mainly in continuous search spaces [35]. Differential evolution was introduced by Storn and Price in 1995 [36]. Global optimization is necessary in fields such as engineering, statistics, and finance, but many practical problems have objective functions that are nonlinear, noisy, noncontinuous, and multidimensional or have many local minima and constraints. Such problems are difficult if not impossible to solve analytically. Differential evolution can be used to find approximate solutions to such problems. Differential evolution also includes genetic algorithms, evolutionary strategies, and evolutionary programming. Differential evolution encodes solutions as vectors and new solution, compared to its parent. If the candidate is better than its parents, it replaces the parent in the population. Differential evolution can be applied in numerical optimization [37, 38].

In this paper, a hybrid evolutionary technique is used in order to solve the k-means problem. The proposed algorithm helps clustering technique to escape from being trapped in local optimum. Our algorithm takes the benefits of both algorithms. Also, in this survey, some standard datasets are used for testing the proposed algorithm. To obtain the best cluster centers, in proposed algorithm, the advantages of BA (bees algorithm) and DE (differential evolution) are used with a data preprocessing technique called CCIA (cluster center initialization algorithm) for data analysis. Through experiments, the proposed CCIA-BADE-K algorithm has shown that this algorithm efficiently selects the exact cluster centers.

The main contribution of this paper is the introduction of a novel combination of evolutionary algorithm according to bees algorithm and differential evolution to overcome data analysis problem and hybrid with CCIA (cluster center initialization algorithm) preprocessing technique.

The rest of this paper is arranged as follows: in Section 2, the data clustering issue is introduced. In Sections 3 and 4, the classic principles of the DE and BA evolutionary algorithm are discussed. In Section 5, the suggested approach is introduced. In Section 6, experimental results of proposed algorithm are shown and compared with PSO-ANT, SA, ACO, GA, ACO-SA, TS, HBMO, PSO, and k-means on benchmark data and finally Section 7 presents the concluding remarks.

2. Data Clustering

Clustering is defined as grouping similar objects either physically or in abstract. The groups inside one cluster have the most similarity with each other and the maximum diversity with other groups' objects [39].

Definition 1. Suppose the set of $X = \{x_1, x_2, \ldots, x_n\}$ containing n objects. The purpose of clustering is to group objects in k clusters as $C = \{c_1, c_2, \ldots, c_k\}$ while each cluster satisfies the following conditions [40]:

(1) $C_1 \cup C_2 \cup \cdots \cup C_k = X$;

(2) $C_i \neq \emptyset, i = 1, \ldots, k$;

(3) $C_i \cap C_j = \emptyset$.

According to the mentioned definition, the possible modes for clustering n objects in k clusters are obtained as follows:

$$\mathrm{NW}(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^i \binom{k}{i}(k-i)^n. \quad (1)$$

(1) **Input**: Data SET ($X = \{x_1, x_2, \ldots, x_n\}$), Attribute Set ($A = \{A_1, A_2, \ldots, A_q\}$), Cluster Number ($K$),
(2) **Output**: Clusters Seed-Set ($SC = \{sc_1, sc_2, \ldots, sc_H\}$)
(3) **Begin**
(4) While ($\forall A_j \in A$)
    (4.1) Compute Standard Deviation (σ_j) and Mean (μ_j)
    (4.2) Compute Cluster Center ($\mathbf{e} = 1, 2, \ldots, k$)
    $X_e = Z_e * \sigma_j + \mu_j \quad Z_e = \dfrac{2 * e - 1}{2 * k}$
    (4.3) Execute k-means on this attribute
    (4.4) Allocate cluster labels obtained from Step (4.3) to every data pattern
(5) Find unique patterns ($H \geq k$) and clustering each data with obtained patterns.
(6) Return SC
(7) End

Algorithm 1: Pseudocode of CCIA algorithm.

In most approaches, the cluster number, that is, k, is specified by an expert. Relation (1) implies that even with a given k, finding the optimum solution for clustering is not so simple. Moreover, the number of possible solutions for clustering with n objects in k clusters increases by the order of $k^n/k!$. So, obtaining the best mode for clustering n objects in k clusters is an intricate NP-complete problem which needs to be settled by optimization approaches [5].

2.1. The K-Means Algorithm. There have been many algorithms suggested for addressing the clustering problem and among them the k-means algorithm which is one of the most famous and most practical algorithms [41]. In this method, besides the input datasets, k samples are introduced into the algorithm as the initial centers of k clusters. These representing k's are usually the first k data samples [39]. The way these k representatives are chosen influences the performance of K-means algorithm [42]. The four stages of this algorithm are shown as follows.

Stage I. Choose k data items randomly from $X = \{x_1, x_2, \ldots, x_n\}$ as cluster centers of $(m_1, m_2, \ldots, m_k)$.

Stage II. Based on relation (2), add every data item to a relevant cluster. For example, if the following relation (2) holds, the object x_i from the set of $X = \{x_1, x_2, \ldots, x_n\}$ is added to the cluster c_j

$$\left\|x_i - m_j\right\| < \left\|x_i - m_p\right\| \quad 1 \leq p \leq k, \ j \neq p. \tag{2}$$

Stage III. Now, based on the clustering of Stage II, the new cluster centers $(m_1^*, m_2^*, \ldots, m_k^*)$ are calculated by using relation (3) as follows (n_i is the number of objects in the cluster i):

$$m_i^* = \frac{1}{n_i} \sum_{x_j \in C_i} x_j \quad 1 \leq i \leq k. \tag{3}$$

Stage IV. If the cluster centers are changed, repeat the algorithm from Stage II, otherwise do the clustering based on the resulted centers.

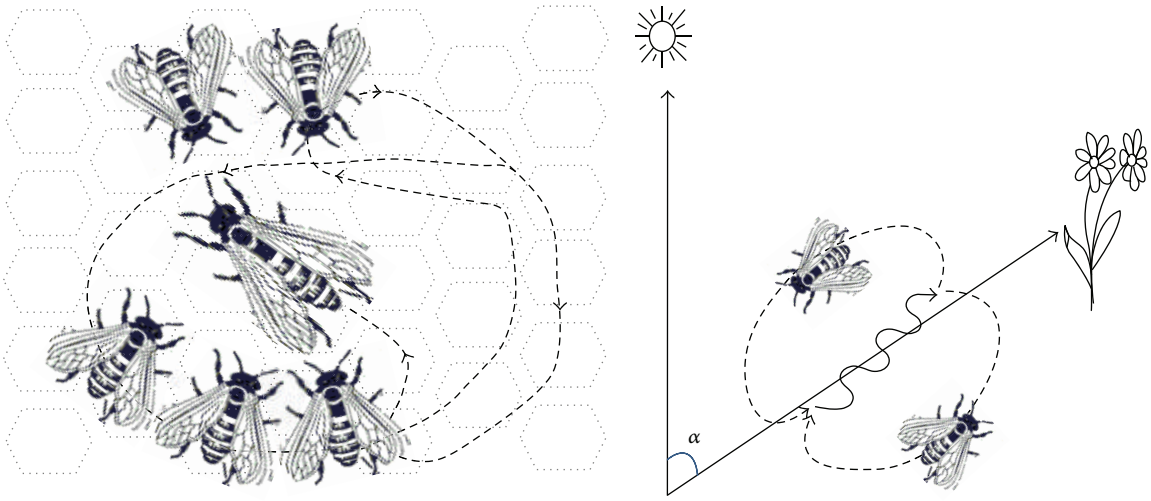
The performance of the k-means clustering algorithm relies on initial centers and this is a major challenge in this algorithm. Random selection of initial cluster centers makes this algorithm yield different results for different runs over the same datasets, which is considered as one of the potential disadvantages of this algorithm [43]. This mix is not sensitive to center initialization, but it still has tendency towards local optimality. In this algorithm, strong ties among data points and the nearest data centers cause cluster centers not to exit from their local dense ranges [44].

The algorithm of bees, first developed by Karaboga and Basturk [3] and Pham et al. in 2006 [30], is a new swarm-based algorithm to search solutions independently. The algorithm was inspired by the behavior of food foraging from swarms of honeybees. In classic edition, the algorithm used random search to find neighborhood to solve optimization problems and issues.

2.2. Algorithm for Finding Cluster Initial Centers. In this study, with regards to efficiency purposes, all data objects are first clustered using k-means algorithm to find the initial cluster centers to be used in the solutions based on all their attributes. Based on the generated clusters, the pattern for an object is produced from each attribute at any stage.

Objects with the same patterns are located in one cluster and hence all objects are clustered. The obtained clusters in this stage will be more than the original number of clusters. For more information, refer to paper [6]. In this paper, clustering is completed in two stages. The first stage is performed as discussed above and in the second stage similar clusters are integrated with each other until achieving a given number of clusters. Algorithm 1 shows the proposed approach for initial clustering of data objects and the achieved cluster centers are called seed points.

As can be observed in Algorithm 1, for every attribute of data objects, a cluster label is generated and this label is added to the data object pattern. Objects with identical patterns are placed in one cluster. To produce data object labels based on each attribute, first the mean and standard deviation of that attribute are computed for all data objects. Thereafter, based on the mean and standard deviation, the range of attribute



(a) Dance floor (b) Dance languages

Figure 1: Waggle dance of the honeybee.

values are broken into k identical intervals so that the tail of each interval appears as an initial cluster center. Thus, using the initial centers, all data objects are clustered by the k-means method.

2.3. Fitness Function. To calculate the fitness of each solution, the distance between the centers of clusters and each data will be used. To do this, first a set of cluster centers will be generated randomly and then clustering of the numerator will be conducted based on (2). Now, according to centers obtained in the interaction step, the new centers of the clusters and fitness of solutions based on (3) will be calculated [40]

$$\text{Fitness}(C) = \sum_{i=1}^{k} \sum_{x_j \in C_i} \left\| x_j - m_i^* \right\|. \quad (4)$$

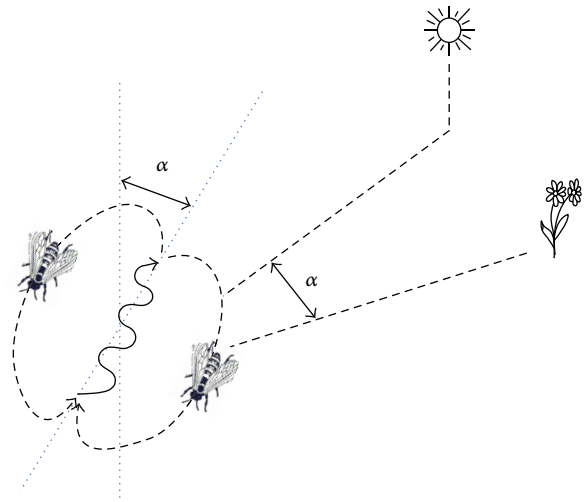


Figure 2: Communication and information sharing.

3. The Dance Language of Bees

For honeybees, finding nectar is essential to survival. Bees lead others to specific sources of food and then scout bees start to identify the visited resources by making movements as "dancing." These dances are very careful and fast in different directions. Dancers try to give information about a food resource by specifying the direction, distance, and quality of the visited food source [45].

3.1. Describing the Dance Language. There are two kind of dance for Observed bees including "round dance" and "waggle dance" [46]. When a food resource is less than fifty meters away, they do round dance and when a food resource is greater than fifty meters away, they perform waggle dance (Figure 1).

There are some concepts in this dance, in which the angle between vertical and waggle run is equal to the angle between the sun and food resource. Dance "tempo" shows the distance of food resource (Figure 2). A slower dance tempo means that a food resource is farther and vice versa [47]. Another concept is the duration of dance and a longer dance duration means that a food resource is rich and better [45]. Audiences are other bees, which follow the dancer. In this algorithm, there are two kinds of bees, SCOUTS are bees that find new food sources and perform the dance. RECRUTTS are bees that follow the scout bees, dance, and then forage. One of the first people that translate the waggle dance mining was Austrian etiologist Karl von Frisch.

Distance between flowers and hive is demonstrated by the duration of the waggle dance. The flowers that are farther from the hive have longer waggle dance duration. Each hundred meters distance between flowers from the hive is shown in the waggle dance phase close to 75 milliseconds.

3.2. Bee in Nature. A colony of honeybees can extend itself over long distances (more than 10 km) and in multiple

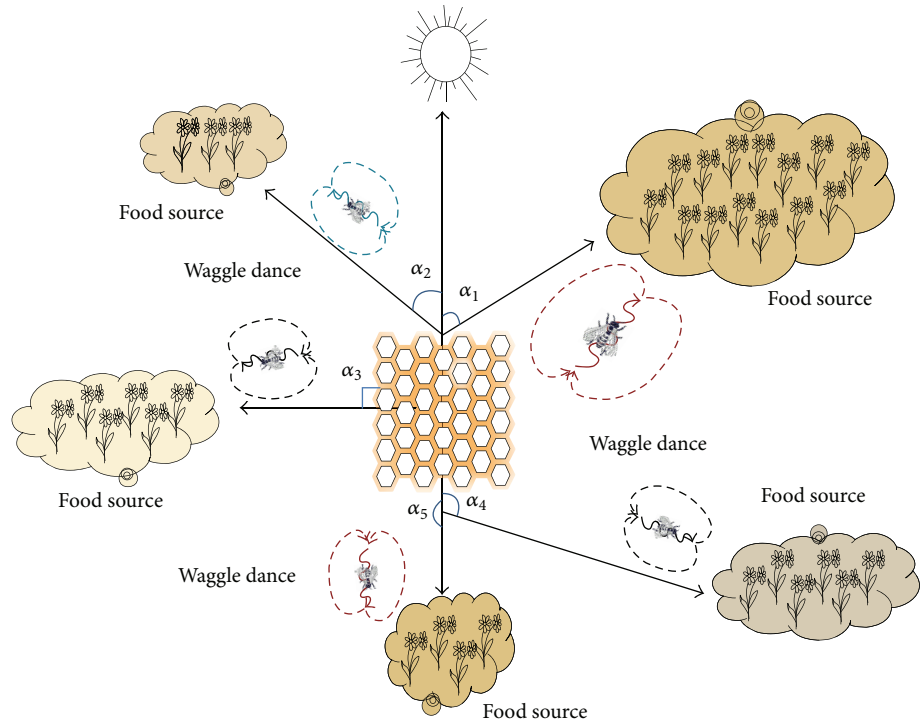


Figure 3: The intelligent foraging behavior of honeybee colony.

(1) Initialize population with random solutions. (n scout bees are placed randomly in the search space.)
(2) Evaluate fitness of the population.
(3) While (Repeat optimization cycles for the specified number)
(4) Select sites for neighborhood search. (Bee that have the highest fitness are chosen as "selected" and sites visited by them are chosen for neighborhood search.)
(5) Recruit bees for selected sites (more bees for best e sites) and evaluate fitness.
(6) Select the fittest bee from each patch. (For each patch, only the bee with the highest fitness will be selected to form the next bee population.)
(7) Assign remaining bees to search randomly and evaluate their fitness.
(8) End While

Algorithm 2: Pseudocode of Basic Bee Algorithm.

directions simultaneously to exploit a large number of food sources. In principle, flower patches with plentiful amount of nectar or pollen that can be collected with less effort should be visited by more bees, whereas, patches with less nectar or pollen should receive fewer bees [35, 47].

The foraging process begins in a colony with the scout bees being sent out to search for promising flower patches. Scout bees move randomly from one patch to another. During the harvest season, a colony continues its exploration, keeping a percentage of the population as scout bees. When the scout bees return to the hive, those that found a patch, which is rated above a certain quality threshold (measured as a combination of some constituents, such as sugar content), deposit their nectar or pollen and go to the "dance floor" to perform a dance known as "waggle dance" [46]. The waggle dance is essential for colony communication and contains three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive, and its quality rating (or fitness). This information helps the colony to send its bees to flower patches precisely, without using guides or maps [45]. After the waggle dance on the dancing floor, the dancers (i.e., scout bee) go back to the flower patch with follower bees that are waiting inside the hive. More follower bees are sent to patches that are most promising [48, 49]. The flowchart of bee algorithm is shown in Figure 4 [50].

The Basic Bee Algorithm is shown as in Algorithm 2 [51].

4. Differential Evolution

Differential evolution is a type of standard genetic algorithm. Differential evolution algorithm evaluates the initial population by using probability motion and observation models and population evolution is performed by using evolution

(1) **Begin**
(2) First j_0 is selected randomly between 1 and D
(3) j_0 is added to set J
(4) For all values of j the following operations are repeated:
    (a) One random number is generated such as rand_j that has uniform distribution between zero and one
    (b) If r_j is less than or equal to P_{cr} then number of j is added to J set
(5) **End**.

Algorithm 3: Pseudocode of the binomial crossover.

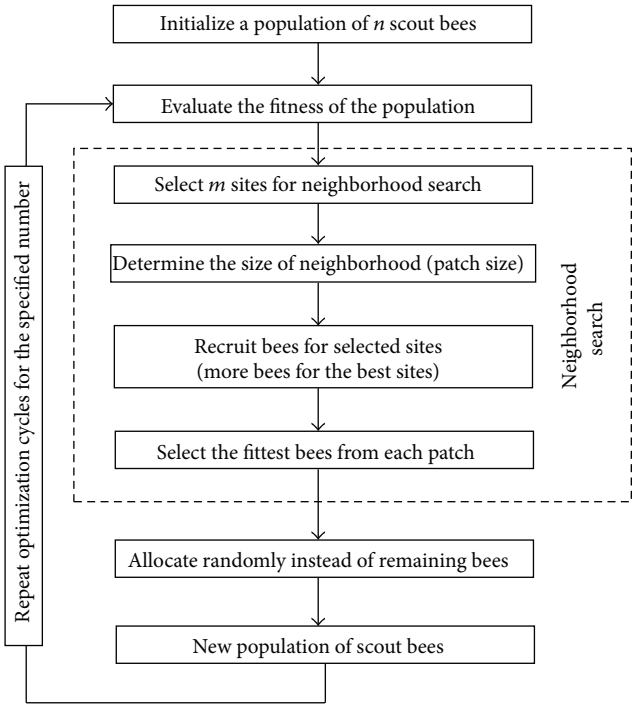


Figure 4: The flowchart of bee algorithm.

operators [52]. The main idea in the differential evolution algorithm is to generate a new solution for each solution by using one constant member and two random members. In each generation, the best member of population is selected and then the difference between each member of population and the best member is calculated. Two random members are then selected and the difference between them is calculated. Coefficient of this difference is added to ith member and thus a new member is created. The cost of each new member is calculated and if the cost value of the new member is less, the ith member is replaced instead of ith member; otherwise, the previous value can be kept in the next generation [35].

Differential evolution is one of the population based popular algorithms that uses point floating (real coded) for presentation as follows [53]:

$$Z_i(t) = \left[Z_{i,1}(t), Z_{i,2}(t), \ldots, Z_{i,D}(t)\right], \tag{5}$$

where t is the number of generation (iteration), i refers to members (population), and d is the number of optimization parameters. Now, in each generation (or each iteration of algorithm) to perform changes on members of population $Z_i(t)$, one donor vector $Y_i(t)$ is formed. The various methods of DE are used to determine how to make the donor vector. The first kind of DE named 1/rand/DE generates ith member $Y_i(t)$, in which three members of current generation (r_1, r_2, r_3) are chosen randomly as

$$r_1 \neq r_2 \neq r_3 \in \{1, 2, \ldots, D\}. \tag{6}$$

Then, the difference between two vectors from three selected vectors are calculated and multiplied by F coefficient and with the third vector added [53]. Therefore, donor vector $Y_i(t)$ is obtained. Calculation process of donor vector for jth element from ith vector can be demonstrated as follows [54]:

$$Y_{i,j}(t) = Z_{r1,j}(t) + F\left(Z_{r2,j}(t) - Z_{r3,j}(t)\right). \tag{7}$$

To increase the exploration of algorithm a crossover operation is then performed. Differential algorithm has generally two kinds of crossover exponential and binomial [55]. In this paper to save time, the binomial mode has been used. To apply the binomial crossover, it requires that set of J is constituted as in Algorithm 3.

Therefore, for each target vector $Z_i(t)$, there is a trial vector as follows [56]:

$$R_{i,j}(t) = \begin{cases} Y_{i,j}(t) & \text{If } \text{rand}_{j,i} \leq P_{\text{cr}} \text{ or } j = j_0 \ \ i = 1, 2, \ldots, N \\ Z_{i,j}(t) & \text{If } \text{rand}_{j,i} > P_{\text{cr}} \text{ and } j \neq j_0 \ \ j = 1, 2, \ldots, D, \end{cases} \tag{8}$$

where j is equal to $j = 1, 2, \ldots, D$ and rand_j is uniform distribution number between $[0, 1]$. Set of J is guaranteed where there is at least one difference between $R_i(t)$ and $Z_i(t)$. In the next step, the selection process is performed between target vector and trial vector as follows:

$$Z_i(t+1) = \begin{cases} R_i(t) & \text{If } f\left(R_i(t)\right) \leq f\left(Z_i(t)\right) \ \ i = 1, 2, \ldots, N \\ Z_i(t) & \text{otherwise,} \end{cases} \tag{9}$$

where $f(\cdot)$ is a function that should be the minimum. In this paper, to escape from premature convergence, two new strategies of merging have been studied. In the basic DE are used difference vector of $Z_{r2,j}(t) - Z_{r3,j}(t)$ multiplied F

(1) Define algorithm parameter
(2) Generate and evaluate initial population or solutions
(3) For all members of population per form the following steps
    (a) With mutation operator create a new trial solution
    (b) By using the crossover generate new solution and evaluate them
    (c) Replace new solution with current solution if new solution is better than current solution otherwise, the current solution is retained.
(4) Return to step three if termination condition is not achieved.

Algorithm 4: Pseudocode of differential evolution.

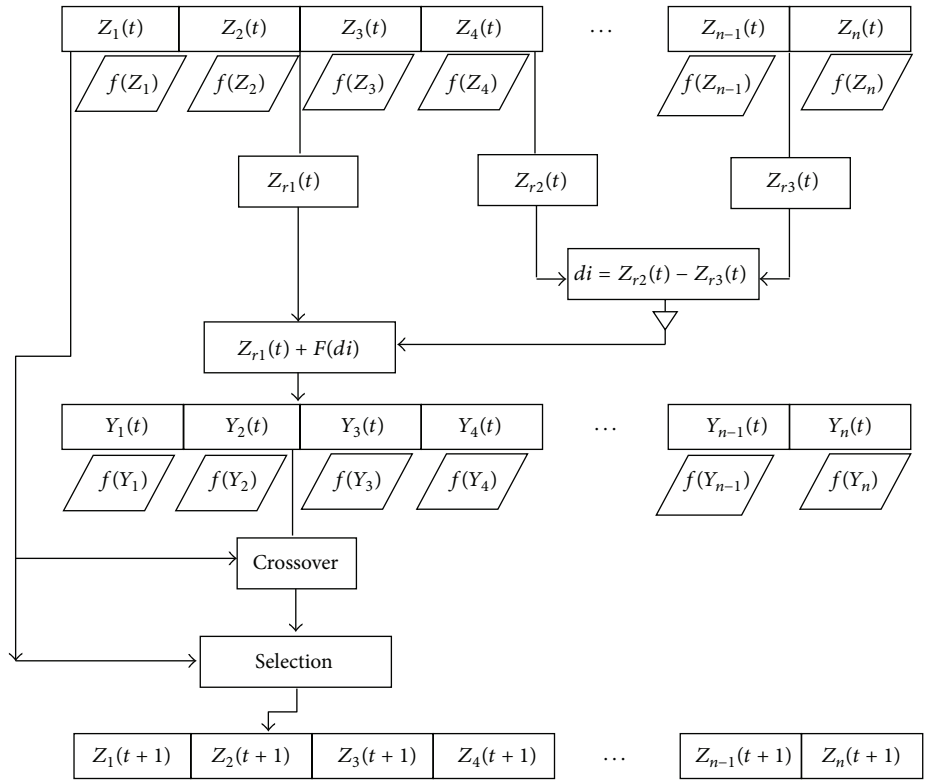


Figure 5: The process of differential evolution.

where F is control parameter between 0.4 and one [55, 57]. To improve the convergence feature in the DE, this paper makes the following proposal:

$$F = 0.5\,(\mathrm{Rnd} + 1)\,, \tag{10}$$

where Rnd is uniform distribution number between zero and one. Generally, the DE algorithm steps are as in Algorithm 4.

In Figure 5, the process of differential evolution is illustrated.

5. Proposed Algorithm

As noted in the former sections, studies conducted on the BA method have shown that this algorithm can be a powerful approach with enough performance to handle different types of nonlinear problems in various fields. However, it can be possibly trapped into local optimum. Lately, several ideas have been used to reduce this problem by hybrid different evolutionary techniques such as partial swarm optimization, genetic algorithm, and simulating annealing. In most population based evolutionary algorithms, in each iteration, new members are generated and then the movement operations are applied to explore new positions based on providing better opportunities. To increase the diversity of algorithm, in the differential evolution algorithm, all members have a possibility to win the global optimum and move to that side. The ability of the best particle to local search also depends on the other particles by selecting the two other particles and calculating the difference between them. This situation may lead to local convergence.

In this proposed algorithm, to escape from random selecting of the global best particle, we used competency selection for choosing the global best particle. If particle is better than the other solutions, then the probability of being selected is greater.

```
Begin
  (a)  Find seed cluster center (preprocessing)
  (b)  Create an initial Bees population randomly with n Scout Bees
  (c)  Calculate the objective function for each individual
  (d)  Sort and update best site ever found
  (e)  Select the elite sites, non-elite sites, and non-selected site (three site groups)
  (f)  Determine number of recruited bees for each kind of site
  (g)  While (iteration < 100)
       (I)   For each selected kind of sites
                 % calculate the neighborhoods
         (1)       For each recruited bees
                       % Mutation
         (2)       Choose target site and base site from this group
         (3)       Random choice of two sites from this group
         (4)       Calculate weighted difference site
         (5)       Add to base selected site
                       % Crossover
         (6)       Perform crossover operation with Crossover Probability
         (7)       Evaluate the trial site that is generated
                       % update site
         (8)       If trial site is less than target site
         (9)           Select trial site instead of target site
         (10)      else
         (11)          Select target site
         (12)      End if
       (II)  End (for of recruited bees)
  (h)    End (for of selected Sites)
  (i)    Sort and update best site ever found
End
```

Algorithm 5: Pseudocode of proposed CCIA-BADE-K algorithm.

The basic idea behind the proposed algorithm is that our solutions are grouped based on the bees' algorithm.

On the other hand, in this algorithm new approach is proposed to the movement and selects the recruiting bees for selecting sites. This algorithm classified the bees into three groups and named them elite sites, nonelite sites, and nonselected site. To increase diversity, the two modes for movement based on the differential evolution algorithm operator as parallel mode and serial mode were used. The suggested algorithm tries to use the advantage of these algorithms to find the best cluster center and to improve simulation results. In other words, in this algorithm, first, a preprocessing technique is performed on the data and then the proposed hybrid algorithm is used to find the best cluster center for k-means problem.

The flowchart and pseudocode of the combined algorithm, called CCIA-BA-DE, are illustrated in Algorithm 5 and Figure 6.

6. Application of CCIA-BA-BE on Clustering

The application of CCIA-BADE-K algorithm on the clustering problem in this section is presented. To perform the CCIA-BADE-K algorithm to find best cluster centers, the following steps should be repeated and taken.

Step 1 (generate the seed cluster center). This step is a preprocessing step to find the seed cluster center to choose the best interval for each cluster.

Step 2 (generate the initial bees' population randomly). In other words, generate initial solutions to find the best cluster centers statistically

$$\text{Population} = \begin{bmatrix} \text{center}_1^T \\ \text{center}_2^T \\ \vdots \\ \text{center}_{n\text{Scout}}^T \end{bmatrix}, \tag{11}$$

where center is a vector with k cluster and each vector has p dimension:

$$\begin{aligned} \text{center}_i &= \left[C_1, C_2, \ldots, C_k\right], \quad i = 1, 2, \ldots, n\text{Scout}, \\ C_j &= \left[c_1, c_2, \ldots, c_p\right], \quad j = 1, 2, \ldots, k, \\ C_j^{\min} &< C_j < C_j^{\max}, \end{aligned} \tag{12}$$

where C_j is cluster center of j for ith scout bee and p is the number of dimension for each cluster center. In fact, each

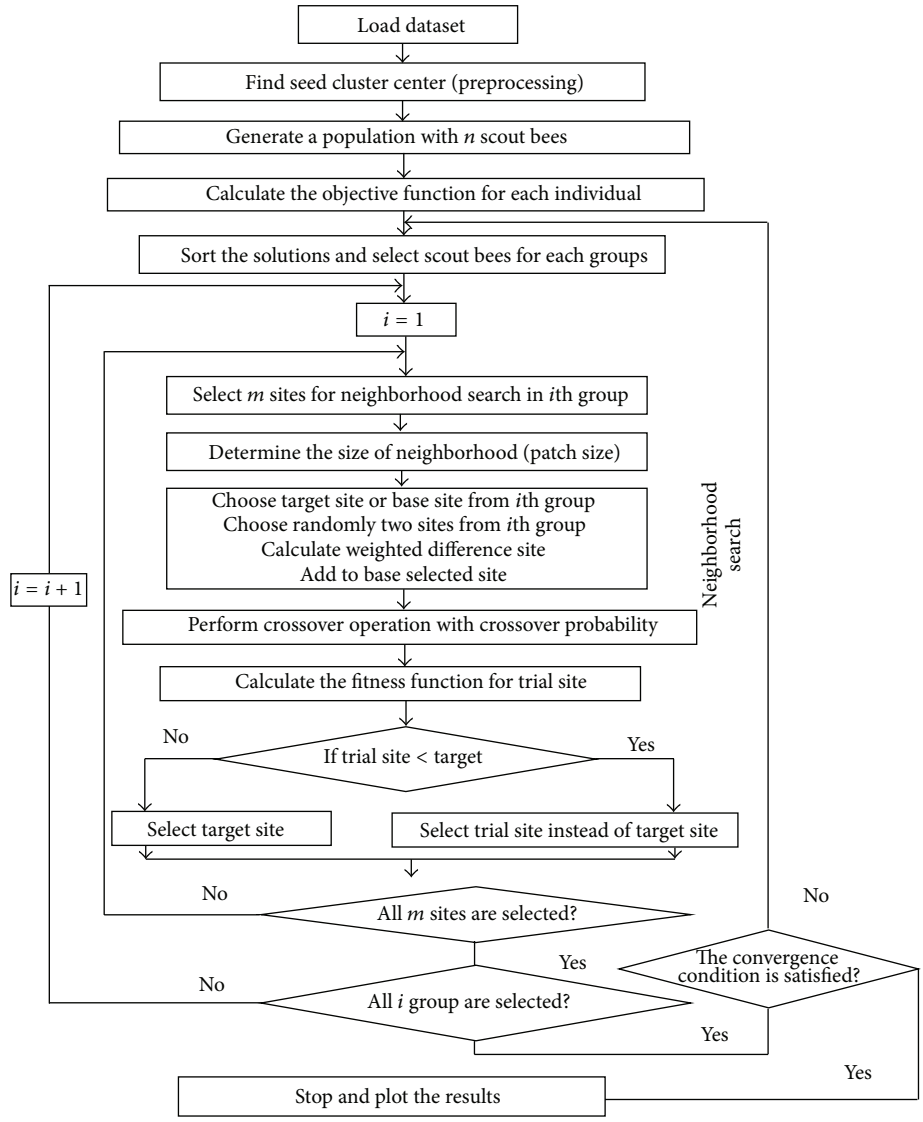


Figure 6: Flowchart of proposed algorithm.

solution in the algorithm is a matrix with $k \times p$. $c_i^{\min}$ and $c_i^{\max}$ are values of the minimum and maximum for each dimension (each feature of center).

Step 3 (calculate the objective function for each individual). Calculate the cost function for each solution (each site) in this algorithm.

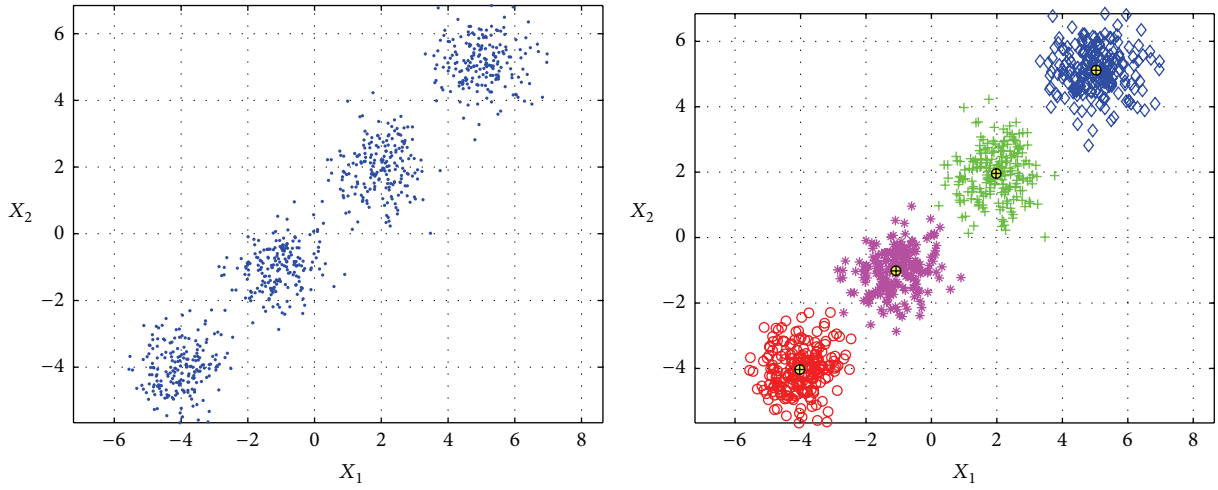
Step 4 (sort the solutions and select scout bees for each groups). The sorting of the site is carried out based on the objective function value.

Step 5 (select the first group of sites). Finding the new solutions is performed by selecting the group of sites. There are three groups of sites in which the first group or elite sites are evaluated to find the neighbors of the selected site followed by nonelite site and finally nonselected sites. To find the neighbors of each group of sites, either the serial mode or parallel mode may be used. This algorithm used parallel model.

Step 6 (select the number of bees for each site). Numbers of bees for each site depend on their group and are considered as competence, more bees for better site. If the site is rich then more bees are allocated to this site. In other words, if the solution is better, it is rated as more important than the other sites.

Step 7 (performing the differential evolution operator (mutation)). In this step, the target site is chosen from the group sites and then two other sites from this group are selected randomly to calculate the weighted difference between them. After calculating this difference, it is added to base trial site as shown in the following equation:

$$v_{i,G+1} = x_{r1,G} + W\left(x_{r2,G} - x_{r3,G}\right), \quad r_1 \neq r_2 \neq r_3 \in \{1, 2, \ldots, N\}, \tag{13}$$



(a) First artificial dataset

(b) Clustered dataset after with cluster centers

FIGURE 7: Used artificial dataset one.

where $x_{r1,G}$ is the target site, W is the weight, and the $x_{r2,G}$, $x_{r3,G}$ are the selected sites from target's group. $v_{i,G+1}$ is the trial solution for comparison purposes.

Step 8 (perform crossover operation with crossover probability). The recombination step incorporates successful solutions from the previous generation. The trial vectors $u_{i,G+1}$ is developed from the elements of the target vector, $x_{i,G}$ and the elements of donor vector $v_{i,G+1}$. Elements of the donor vector enter the trial vector with probability CR

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{If } \text{rand}_{j,i} \le \text{CR or } j = I_{\text{rand}} \quad i = 1, 2, \ldots, N \\ x_{j,i,G} & \text{If } \text{rand}_{j,i} > \text{CR and } j \ne I_{\text{rand}} \quad j = 1, 2, \ldots, L, \end{cases} \tag{14}$$

where $\text{rand}_{j,i} \sim U[0, 1]$ and I_{rand} is a random integer from $[1, 2, \ldots, L]$ and I_{rand} ensures that $u_{i,G+1} \ne x_{i,G}$.

Step 9 (calculate the cost function for trial site). In the selection step, target vector $x_{i,G}$ is compared with trial vector $u_{i,G+1}$. There are two modes to calculate the new site as follows:

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{If } f(u_{i,G+1}) \le f(x_{i,G}) \quad i = 1, 2, \ldots, N \\ x_{i,G} & \text{otherwise.} \end{cases} \tag{15}$$

Trial vector $u_i(G)$ is compared to target vector $x_i(G)$. To use greedy criterion, if $u_i(G)$ is better than the $x_i(G)$, then replace $x_i(G)$ with $u_i(G)$; otherwise, $x_i(G)$ "survive" and $u_i(G)$ are discarded.

Step 10. If not all sites from this group are selected, go to Step 6 and select another site from this group; otherwise, go to the next step.

TABLE 1: Table type styles.

Dataset name	Dataset attribute		
	Dataset size	Cluster number	Attribute number
Iris	150 (50, 50, 50)	3	4
Wine	178 (59, 71, 48)	3	13
CMC	1473 (629, 334, 510)	3	9
Glass	214 (70, 17, 76, 13, 9, 29)	6	9

Step 11. If not all groups are selected, go to Step 5 and select the next group; otherwise, go to the next step.

Step 12 (check the termination criteria). If the current number of iteration does not reach the maximum number of iterations, go to Step 4 and start next generation; otherwise, go to the next step.

7. Evaluation

To evaluate the accuracy and efficiency of the proposed algorithm, experiments have been performed on two artificial datasets, four real-life datasets and four standard datasets to determine the correctness of clustering algorithms. This collection includes Iris, Glass, Wine, and Contraceptive Method Choice (CMC) datasets that have been chosen from standard UCI dataset.

The suggested algorithm is coded by an appropriate programming language and is run on an i5 computer with 2.60 GHz microprocessor speed and 4 GB main memory. For measuring the performance of the proposed algorithm, the benchmarks data items of Table 1 are used.

The execution results of the proposed algorithm over the selected datasets as well as the comparison figures relative to K-means, PSO, and K-NM-PSO results in [10] are tabulated in Table 2. As easily seen in Table 2, the suggested

Table 2: The obtained results from implementing the suggested algorithm over selected datasets.

Dataset	K-means [10]	PSO [10]	K-NM-PSO [10]	Proposed Alg.	
				Result	CPU
				Time (S)	
Iris	97.33	96.66	96.66	96.5403	~15
Wine	16555.68	16294.00	16292.00	16,292.25	~30
CMC	5542.20	5538.50	5532.40	5532.22	~57
Glass	215.68	271.29	199.68	210.4318	~34

Table 3: The results of implementing the algorithms over Iris test data for 100 runs.

Method	Result			CPU time (S)
	Best	Average	Worst	
PSO-ACO-K	96.650	96.650	96.650	~16
PSO-ACO	96.654	96.654	96.674	~17
PSO	96.8942	97.232	97.897	~30
SA	97.457	99.957	102.01	~32
TS	97.365	97.868	98.569	~135
GA	113.986	125.197	139.778	~140
ACO	97.100	97.171	97.808	~75
HBMO	96.752	96.953	97.757	~82
PSO_SA	96.66	96.67	96.678	~17
ACO-SA	96.660	96.731	96.863	~25
k-Means	97.333	106.05	120.45	0.4
MY proposed ALG.	96.5403	96.5412	96.5438	~15

Table 4: The results of implementing the algorithms over Wine test data for 100 runs.

Method	Result			CPU time (S)
	Best	Average	Worst	
PSO-ACO-K	16,295.31	16,295.31	16,295.31	~30
PSO-ACO	16,295.34	16,295.92	16,297.93	~33
PSO	16,345.96	16,417.47	16,562.31	~123
SA	16,473.48	17,521.09	18,083.25	~129
TS	16,666.22	16,785.45	16,837.53	~140
GA	16,530.53	16,530.53	16,530.53	~170
ACO	16,530.53	16,530.53	16,530.53	~121
HBMO	16,357.28	16,357.28	16,357.28	~40
PSO_SA	16,295.86	16,296.00	16,296.10	~38
ACO-SA	16,298.62	16,310.28	16,322.43	~84
k-Means	16,555.68	18,061.01	18,563.12	0.7
MY proposed ALG.	16,292.25	16,293.76	16,294.98	~30

Table 5: The results of implementing the algorithms over CMC test data for 100 runs.

Method	Result			CPU time (S)
	Best	Average	Worst	
PSO-ACO-K	5,694.28	5,694.28	5,694.28	~31
PSO-ACO	5,694.51	5,694.92	5,697.42	~135
PSO	5,700.98	5,820.96	5,923.24	~131
SA	5,849.03	5,893.48	5,966.94	~150
TS	5,885.06	5,993.59	5,999.80	~155
GA	5,705,63	5,756.59	5,812.64	~160
ACO	5,701.92	5,819.13	5,912.43	~127
HBMO	5,699.26	5,713.98	5,725.35	~123
PSO_SA	5,696.05	5,698.69	5,701.81	~73
ACO-SA	5,696.60	5,698.26	5,700.26	~89
k-Means	5,842.20	5,893.60	5,934.43	0.5
MY proposed ALG.	5,532.22	5,532.45	5,532.85	~57

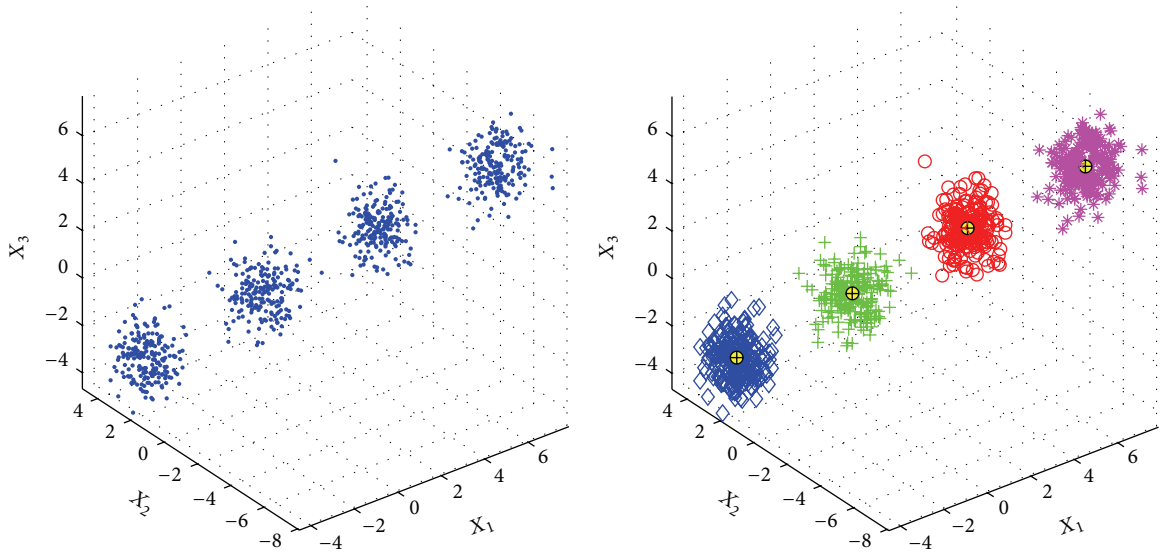
Table 6: The results of implementing the algorithms over Glass test data for 100 runs.

Method	Result			CPU time (S)
	Best	Average	Worst	
PSO-ACO-K	199.53	199.53	199.53	~31
PSO-ACO	199.57	199.61	200.01	~35
PSO	270.57	275.71	283.52	~400
SA	275.16	282.19	287.18	~410
TS	279.87	283.79	286.47	~410
GA	278.37	282.32	286.77	~410
ACO	269.72	273.46	280.08	~395
HBMO	245.73	247.71	249.54	~390
PSO_SA	200.14	201.45	202.45	~38
ACO-SA	200.71	201.89	202.76	~49
k-Means	215.74	235.5	255.38	~1
MY proposed ALG.	210.431	215.54	216.93	~34

algorithm provides superior results relative to K-means and PSO algorithms. The real-life datasets compared with several optimization algorithms are included.

For better study and analysis of the proposed approach, the execution results of the proposed approach along with HBMO, PSO, ACO-SA, PSO-ACO, ACO, PSO-SA, TS, GA, SA, and k-means clustering algorithm results as reported in [8] are tabulated in Tables 3–6. It is worth mentioning that the investigated algorithms of [8] are implemented with MATLAB 7.1, using a Pentium IV system of 2.8 GHz CPU speed and 512 MB main memory.

Frist artificial dataset includes ($n = 800$, $k = 4$, $d = 2$) where n is the number of instance, k is the number of clusters, and d is the number of dimensions. The instances were drawn



(a) Second artificial dataset (b) Clustered dataset after and with cluster centers

FIGURE 8: Used artificial dataset two.

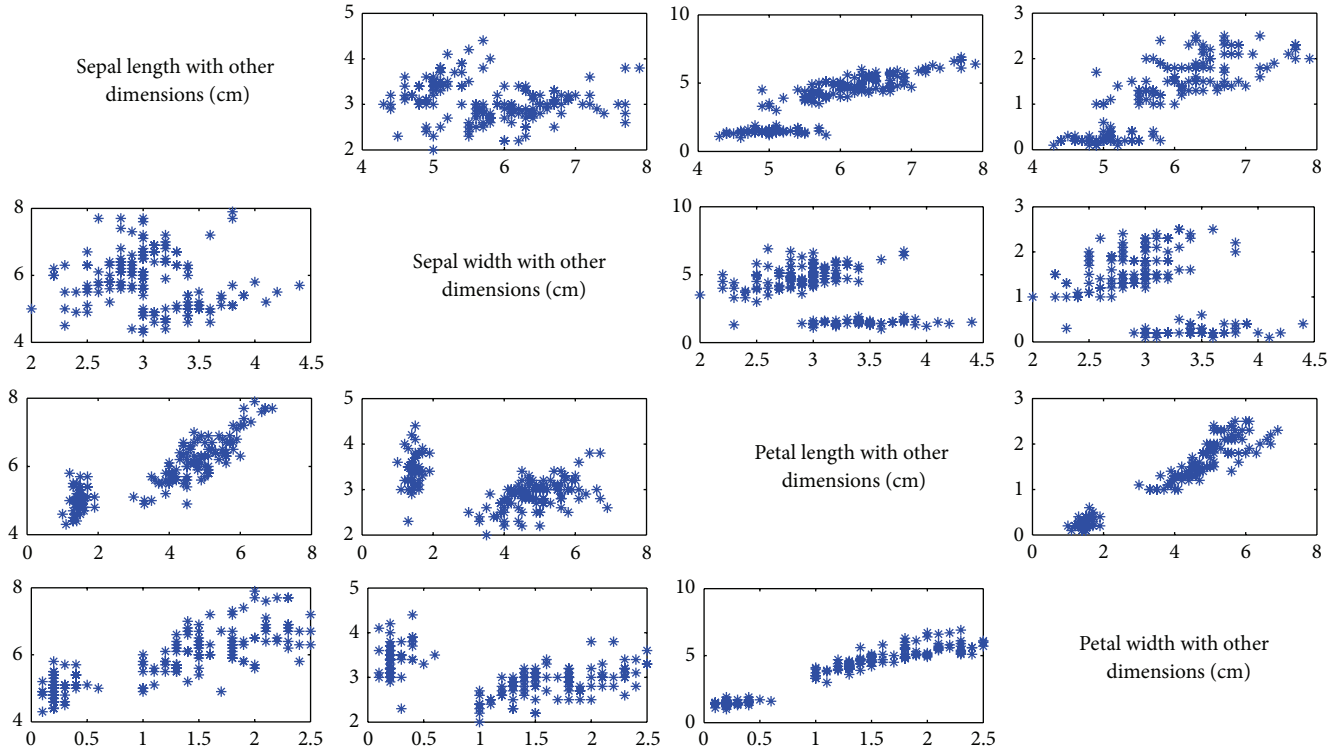


FIGURE 9: The scatter plot to show nonlinear relationship between variables for Iris dataset.

for four absolute classes where each of these groups was distributed as

$$\mathrm{Art}_1\left(\mu = \begin{pmatrix} m_i \\ m_i \end{pmatrix}, \Sigma = \begin{bmatrix} 0.5 & 0.05 \\ 0.05 & 0.5 \end{bmatrix}\right) \quad i = 1, 2, 3, 4 \tag{16}$$

$$m_1 = -4, \quad m_2 = -1, \quad m_3 = 2, \quad m_4 = 5,$$

where Σ and μ are covariance matrix and vector, respectively [10]. The first artificial dataset is demonstrated in Figure 7(a). Figure 7(b) illustrated the clustered data after applying CCIA-BADE-K algorithm on data.

Second artificial dataset includes ($n = 800$, $k = 4$, $d = 3$) where n is the number of instance, k is the number of clusters, and d is the number of dimensions. The instances were drawn

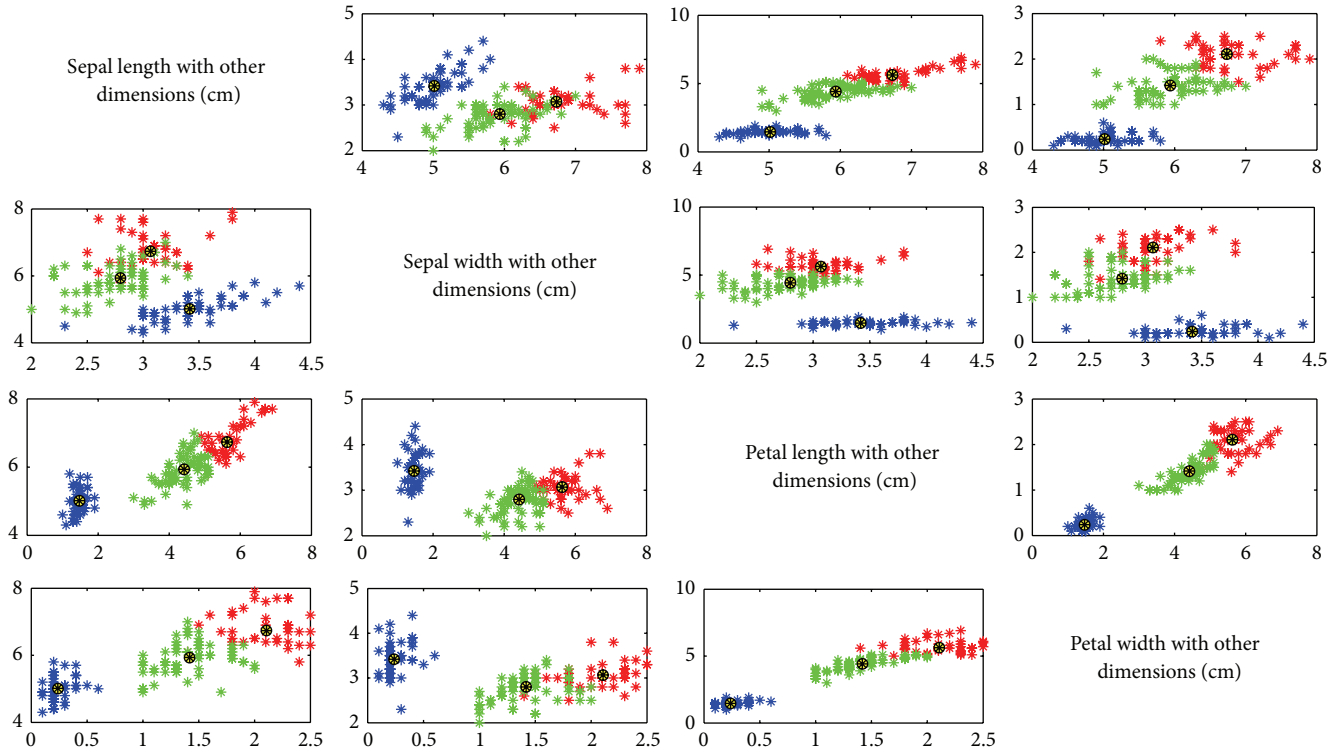


Figure 10: Clustered scatter plot to show nonlinear relationship between variables for Iris dataset.

for four absolute classes where each of these groups was distributed as

$$\mathrm{Art}_2\left(\mu = \begin{pmatrix} m_i \\ -m_i \\ m_i \end{pmatrix}, \Sigma = \begin{bmatrix} 0.5 & 0.05 & 0.05 \\ 0.05 & 0.5 & 0.05 \\ 0.05 & 0.05 & 0.5 \end{bmatrix}\right)$$
$$i = 1, 2, 3, 4 \quad m_1 = -3, \quad m_2 = 0, \quad m_3 = 3, \quad m_4 = 6, \tag{17}$$

where Σ and μ are covariance matrix and vector, respectively [10]. The second artificial dataset is demonstrated in Figure 8. Figure 8 shows clusters after applying proposed algorithm on the artificial dataset.

In Tables 3–6, best, worst, and average results are reported for 100 runs, respectively. The resulting figures represent the distance of every data from the cluster center to which it belongs and is computed by using relation (4). As observed in the table, regarding the execution time, the proposed algorithm generates acceptable solutions.

To clarify the issue, in Figure 10, the scatterplot (scatter-graph) is illustrated. The scatter-graph is one kind of mathematic diagram, which shows the values for a dataset for two variables using Cartesian coordinates. In this diagram, data is demonstrated as a set of spots. This type of diagram is known as a scatter diagram or scatter-gram. This kind of diagram is also used to display relation between response variables with control variables when a variable is below the control of the experimenter. One of the strongest aspects of the scatter-diagram is the ability to show nonlinear relationship between variables. In Figure 9, the scatter-diagram of Iris dataset is displayed and in Figure 10 the clustered Iris data on the scatter-diagram is shown.

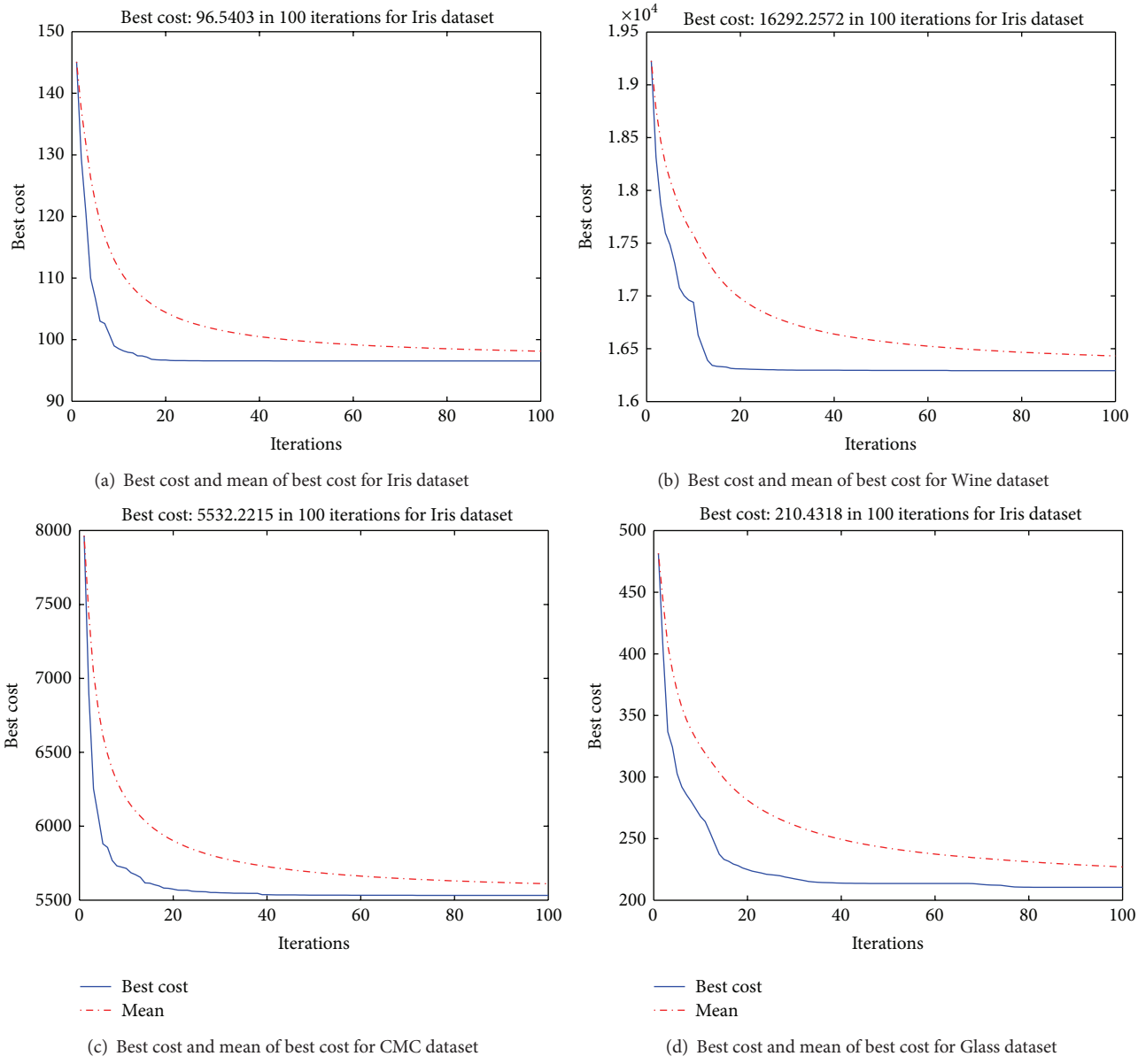
In Table 4, best, worst, and average results of Wine dataset are reported for 100 runs. The resulting figures represent the distance of every data from the cluster center.

In Figure 11 best cost and average best costs of results for all datasets are reported for 100 runs. The resulting figures represent the distance of every data from the cluster center by using relation (4). Figure 11(a) is related to the best cost and mean of best cost for Iris dataset, and Figure 11(b) illustrated the best cost and mean of best cost for Wine dataset. Figure 11(c) reported best cost and mean of best cost for CMC datasets, and finally Figure 11(d) demonstrated mean value of best cost and best cost of Glass dataset.

According to the reported results in Tables 3 to 6, the proposed method over Iris, CMC, and Wine Datasets provides the best results in comparison with other mentioned algorithms. According to Table 6, the suggested algorithm over Glass dataset provides more acceptable results than the alternative algorithms. The reason for this behavior is justified by the fact that as data objects increase in number the efficiency of the alternative algorithms decreases while the deficiency of the suggested algorithm highlights more.

8. Image Segmentation

In Section 7, it was shown that the proposed CCIA-BADE-K algorithm is one of the best methods for data clustering. For further investigation of the performance of algorithm,



(a) Best cost and mean of best cost for Iris dataset

(b) Best cost and mean of best cost for Wine dataset

(c) Best cost and mean of best cost for CMC dataset

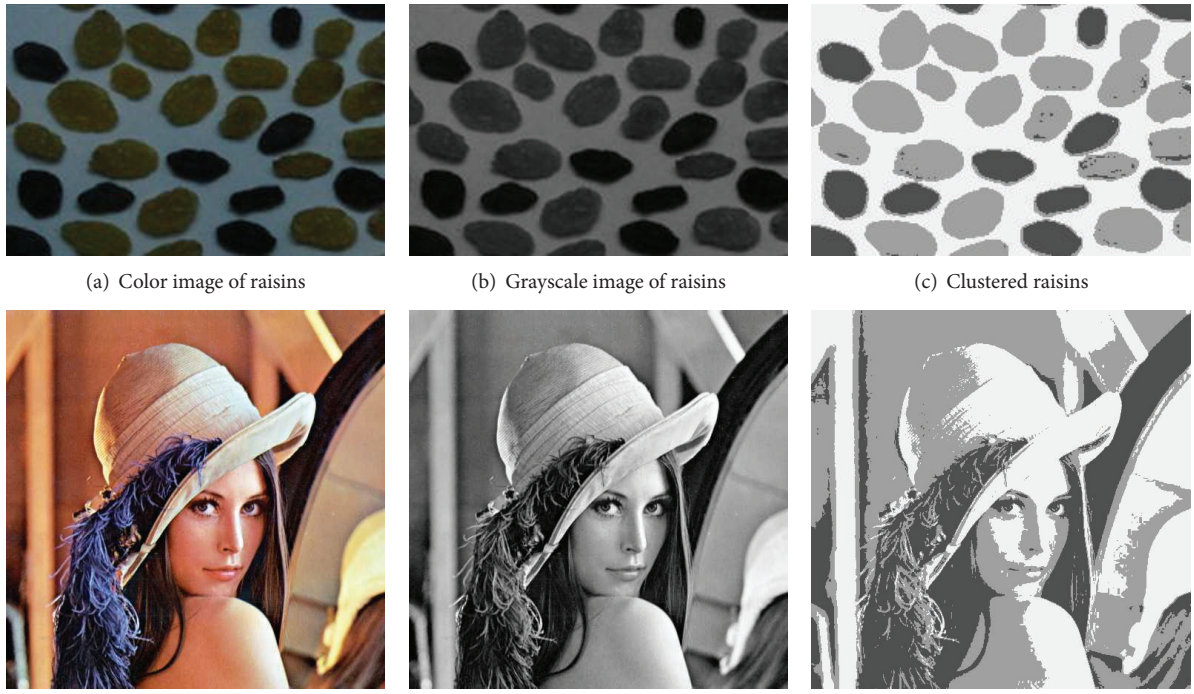
(d) Best cost and mean of best cost for Glass dataset

Figure 11: Best cost and mean of best cost in 100 iterations.

the algorithm was tested on one standard image and one industrial image. Each digital image in RGB space is formed by three-color components consisting of red, green, and blue. Each of these three alone is a grayscale image and the numerical value of each pixel is between 1 and 255. Image histogram is a chart that is made by the number of pixels on an image that is determined based on the brightness level [58]. To obtain a histogram of image it is enough to scroll the whole pixel of image and to calculate the number of pixels for each brightness level. The normalized histogram is obtained by dividing the total number of histogram value to each value of pixels. Normalizing the histogram causes the histogram value to be in $[0, 1]$ interval. Figures 12 and 13 show that image samples in this paper are shown for image segmentation. In Figure 12, the color, grayscale, and clustered modes of these images are shown and, in Figure 13, histogram diagrams for these four images are shown. Furthermore, these segmentation charts will be used to detect segmentation an image.

9. Concluding Remarks

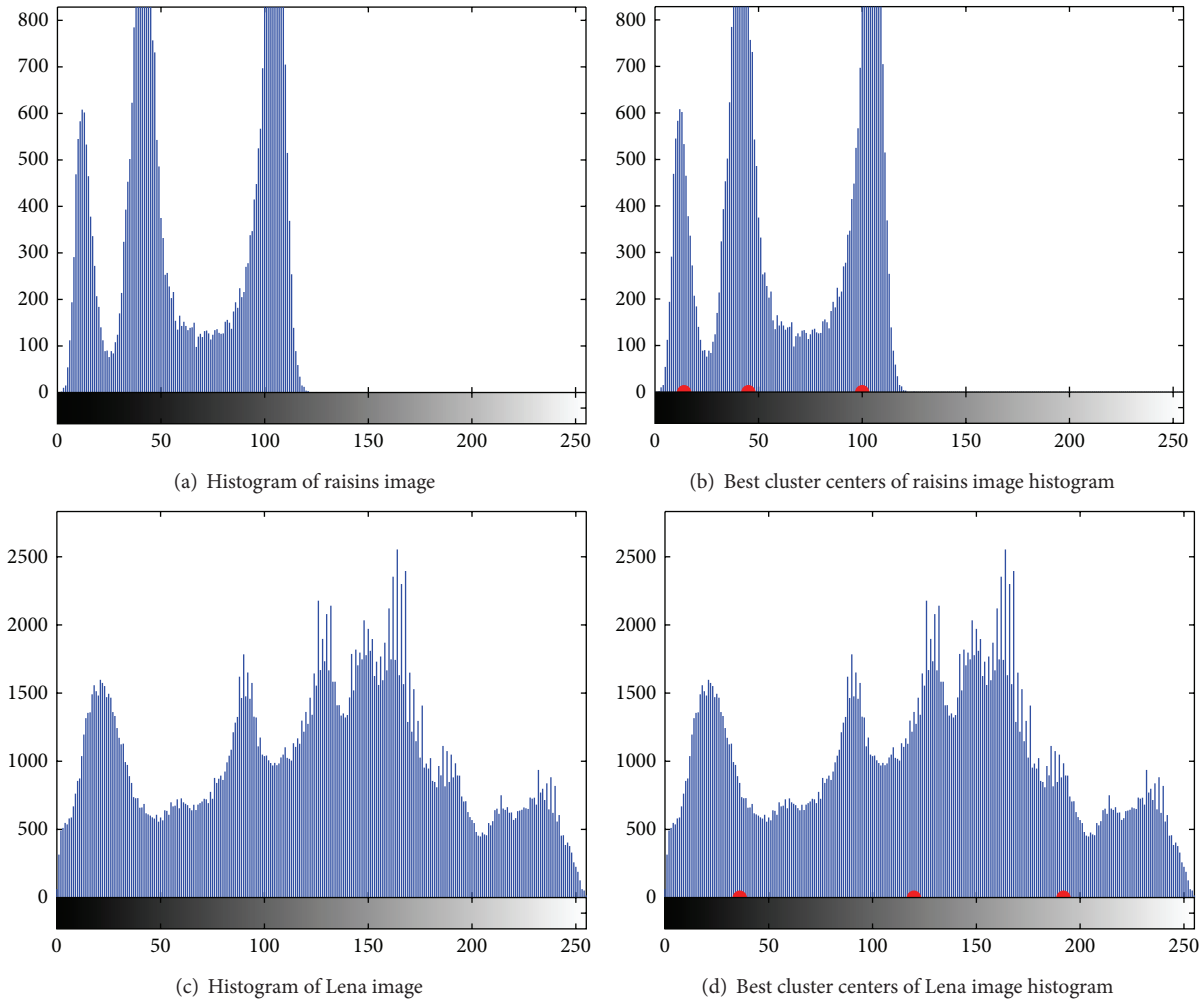
In this paper, a new technique based on a combination of bees algorithm and differential evolution algorithm with k-means was presented. In the proposed algorithm, bee algorithm was assigned to perform globally and differential evolution algorithm was assigned to implement local searching on k-means problem, which is responsible for the task of finding the best cluster centers. The new proposed algorithm CCIA-BADE-K applies abilities of both algorithms and, by removing shortcomings of each algorithm, it tries to use its own strengths to cover other algorithm defects as well



(a) Color image of raisins (b) Grayscale image of raisins (c) Clustered raisins

(d) Color image of Lena (e) Grayscale image of Lena (f) Clustered Lena

Figure 12: Images used for image segmentation with proposed algorithm.



(a) Histogram of raisins image (b) Best cluster centers of raisins image histogram

(c) Histogram of Lena image (d) Best cluster centers of Lena image histogram

Figure 13: Sample used images for image segmentation in the grayscale mode.

as to find best cluster centers that is the proposed seed cluster center algorithm. Experimental results showed that the CCIA-BADE-K algorithm enjoys acceptable results.

Conflict of Interests

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

Acknowledgments

The authors would like to express their cordial thanks to the Ministry of Education (MoE), University Technology Malaysia (UTM), for the Research University Grant no. Q.J130000.2528.06H90. The authors are also grateful to Soft Computing Research Group (SCRG) for their support and incisive comments in making this study a success.

References

[1] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, vol. 20, SIAM, 2007.

[2] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2006.

[3] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[4] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, 2004.

[5] S. Bandyopadhyay and U. Maulik, "An evolutionary technique based on K-means algorithm for optimal clustering in $\mathbb{R}^N$," *Information Sciences*, vol. 146, no. 1–4, pp. 221–237, 2002.

[6] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-means clustering," *Pattern Recognition Letters*, vol. 25, no. 11, pp. 1293–1302, 2004.

[7] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM '02)*, pp. 600–607, McLean, Va, USA, November 2002.

[8] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 183–197, 2010.

[9] C. D. Nguyen and K. J. Cios, "GAKREM: a novel hybrid clustering algorithm," *Information Sciences*, vol. 178, no. 22, pp. 4205–4227, 2008.

[10] Y.-T. Kao, E. Zahara, and I.-W. Kao, "A hybridized approach to data clustering," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1754–1762, 2008.

[11] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 29, no. 3, pp. 433–439, 1999.

[12] K. R. Žalik, "An efficient k$\prime$-means clustering algorithm," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1385–1391, 2008.

[13] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455–1465, 2000.

[14] M. Laszlo and S. Mukherjee, "A genetic algorithm that exchanges neighboring centers for *k*-means clustering," *Pattern Recognition Letters*, vol. 28, no. 16, pp. 2359–2366, 2007.

[15] M. Fathian and B. Amiri, "A honeybee-mating approach for cluster analysis," *The International Journal of Advanced Manufacturing Technology*, vol. 38, no. 7-8, pp. 809–821, 2008.

[16] A. Afshar, O. Bozorg Haddad, M. A. Mariño, and B. J. Adams, "Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation," *Journal of the Franklin Institute*, vol. 344, no. 5, pp. 452–462, 2007.

[17] M. Fathian, B. Amiri, and A. Maroosi, "Application of honey-bee mating optimization algorithm on clustering," *Applied Mathematics and Computation*, vol. 190, no. 2, pp. 1502–1513, 2007.

[18] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187–195, 2004.

[19] T. Niknam, B. B. Firouzi, and M. Nayeripour, "An efficient hybrid evolutionary algorithm for cluster analysis," *World Applied Sciences Journal*, vol. 4, no. 2, pp. 300–307, 2008.

[20] M. K. Ng and J. C. Wong, "Clustering categorical data sets using tabu search techniques," *Pattern Recognition*, vol. 35, no. 12, pp. 2783–2790, 2002.

[21] C. S. Sung and H. W. Jin, "A tabu-search-based heuristic for clustering," *Pattern Recognition*, vol. 33, no. 5, pp. 849–858, 2000.

[22] T. Niknam, B. Amiri, J. Olamaei, and A. Arefi, "An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering," *Journal of Zhejiang University SCIENCE A*, vol. 10, no. 4, pp. 512–519, 2009.

[23] T. Niknam, "An efficient hybrid evolutionary algorithm based on PSO and HBMO algorithms for multi-objective Distribution Feeder Reconfiguration," *Energy Conversion and Management*, vol. 50, no. 8, pp. 2074–2082, 2009.

[24] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.

[25] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[26] W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 459796, 16 pages, 2010.

[27] H. Narasimhan, "Parallel artificial bee colony (PABC) algorithm," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NABIC '09)*, pp. 306–311, IEEE, Coimbatore, India, December 2009.

[28] D. Teodorović, "Bee colony optimization (BCO)," in *Innovations in Swarm Intelligence*, C. Lim, L. Jain, and S. Dehuri, Eds., vol. 248, pp. 39–60, Springer, Berlin, Germany, 2009.

[29] D. Teodorovic, P. Lucic, G. Markovic, and M. Dell' Orco, "Bee colony optimization: principles and applications," in *Proceedings of the 8th Seminar on Neural Network Applications in Electrical Engineering (NEUREL '06)*, pp. 151–156, 2006.

[30] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm—a novel tool for complex optimisation problems," in *Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems (IPROMS '06)*, pp. 454–459, 2006.

[31] R. Akbari, A. Mohammadi, and K. Ziarati, "A novel bee swarm optimization algorithm for numerical function optimization," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 10, pp. 3142–3155, 2010.

[32] H. Drias, S. Sadeg, and S. Yahi, "Cooperative bees swarm for solving the maximum weighted satisfiability problem," in *Computational Intelligence and Bioinspired Systems*, J. Cabestany, A. Prieto, and F. Sandoval, Eds., vol. 3512 of *Lecture Notes in Computer Science*, pp. 318–325, Springer, Berlin, Germany, 2005.

[33] C. Yang, J. Chen, and X. Tu, "Algorithm of fast marriage in honey bees optimization and convergence analysis," in *Proceedings of the IEEE International Conference on Automation and Logistics (ICAL '07)*, pp. 1794–1799, Jinan, China, August 2007.

[34] M. T. Vakil-Baghmisheh and M. Salim, "A modified fast marriage in honey bee optimization algorithm," in *Proceedings of the 5th International Symposium on Telecommunications (IST '10)*, pp. 950–955, December 2010.

[35] 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.

[36] R. Storn and K. Price, *Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, ICSI, Berkeley, Calif, USA, 1995.

[37] V. Feoktistov, "Differential evolution," in *Differential Evolution*, vol. 5, pp. 1–24, Springer, New York, NY, USA, 2006.

[38] K. V. Price, R. M. Storn, and J. A. Lampinen, "The differential evolution algorithm," in *Differential Evolution*, pp. 37–134, Springer, Berlin, Germany, 2005.

[39] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[40] R. J. Kuo, E. Suryani, and A. Yasid, "Automatic clustering combining differential evolution algorithm and k-means algorithm," in *Proceedings of the Institute of Industrial Engineers Asian Conference 2013*, Y.-K. Lin, Y.-C. Tsao, and S.-W. Lin, Eds., pp. 1207–1215, Springer, Singapore, 2013.

[41] W. Kwedlo, "A clustering method combining differential evolution with the K-means algorithm," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1613–1621, 2011.

[42] Y.-J. Wang, J.-S. Zhang, and G.-Y. Zhang, "A dynamic clustering based differential evolution algorithm for global optimization," *European Journal of Operational Research*, vol. 183, no. 1, pp. 56–73, 2007.

[43] M. Babrdelbonab, S. Z. M. H. M. Hashim, and N. E. N. Bazin, "Data analysis by combining the modified k-means and imperialist competitive algorithm," *Jurnal Teknologi*, vol. 70, no. 5, 2014.

[44] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping Multidimensional Data*, J. Kogan, C. Nicholas, and M. Teboulle, Eds., pp. 25–71, Springer, Berlin, Germany, 2006.

[45] J. R. Riley, U. Greggers, A. D. Smith, D. R. Reynolds, and R. Menzel, "The flight paths of honeybees recruited by the waggle dance," *Nature*, vol. 435, no. 7039, pp. 205–207, 2005.

[46] C. Grüter, M. S. Balbuena, and W. M. Farina, "Informational conflicts created by the waggle dance," *Proceedings of the Royal Society B: Biological Sciences*, vol. 275, no. 1640, pp. 1321–1327, 2008.

[47] A. Dornhaus and L. Chittka, "Why do honey bees dance?" *Behavioral Ecology and Sociobiology*, vol. 55, no. 4, pp. 395–401, 2004.

[48] K. O. Jones and A. Bouffet, "Comparison of bees algorithm, ant colony optimisation and particle swarm optimisation for PID controller tuning," in *Proceedings of the 9th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing (CompSysTech '08)*, Gabrovo, Bulgaria, June 2008.

[49] D. T. Pham and M. Kalyoncu, "Optimisation of a fuzzy logic controller for a flexible single-link robot arm using the Bees Algorithm," in *Proceedimgs of the 7th IEEE International Conference on Industrial Informatics (INDIN '09)*, pp. 475–480, Cardiff, Wales, June 2009.

[50] D. T. Pham, S. Otri, A. Ghanbarzadeh, and E. Koc, "Application of the bees algorithm to the training of learning vector quantisation networks for control chart pattern recognition," in *Proceedings of the 2nd Information and Communication Technologies (ICTTA '06)*, vol. 1, pp. 1624–1629, Damascus, Syria, 2006.

[51] L. Özbakir, A. Baykasoğlu, and P. Tapkan, "Bees algorithm for generalized assignment problem," *Applied Mathematics and Computation*, vol. 215, no. 11, pp. 3782–3795, 2010.

[52] P. Rocca, G. Oliveri, and A. Massa, "Differential evolution as applied to electromagnetics," *IEEE Antennas and Propagation Magazine*, vol. 53, no. 1, pp. 38–49, 2011.

[53] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 1679–1696, 2011.

[54] R. Storn, "On the usage of differential evolution for function optimization," in *Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS '96)*, pp. 519–523, June 1996.

[55] U. K. Chakraborty, *Advances in Differential Evolution*, Springer, Berlin, Germany, 2008.

[56] G. Liu, Y. Li, X. Nie, and H. Zheng, "A novel clustering-based differential evolution with 2 multi-parent crossovers for global optimization," *Applied Soft Computing Journal*, vol. 12, no. 2, pp. 663–681, 2012.

[57] Z. Cai, W. Gong, C. X. Ling, and H. Zhang, "A clustering-based differential evolution for global optimization," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 1363–1379, 2011.

[58] M. Abbasgholipour, M. Omid, A. Keyhani, and S. S. Mohtasebi, "Color image segmentation with genetic algorithm in a raisin sorting system based on machine vision in variable conditions," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3671–3678, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 152570, 6 pages
http://dx.doi.org/10.1155/2015/152570

Research Article

Multiple Sparse Measurement Gradient Reconstruction Algorithm for DOA Estimation in Compressed Sensing

Weijian Si, Xinggen Qu, Yilin Jiang, and Tao Chen

Department of Information and Communication Engineering, Harbin Engineering University, 150001 Harbin, China

Correspondence should be addressed to Yilin Jiang; jiangyilin@hrbeu.edu.cn

Received 8 July 2014; Revised 9 November 2014; Accepted 16 March 2015

Academic Editor: Dane Quinn

A novel direction of arrival (DOA) estimation method in compressed sensing (CS) is proposed, in which the DOA estimation problem is cast as the joint sparse reconstruction from multiple measurement vectors (MMV). The proposed method is derived through transforming quadratically constrained linear programming (QCLP) into unconstrained convex optimization which overcomes the drawback that l_1-norm is nondifferentiable when sparse sources are reconstructed by minimizing l_1-norm. The convergence rate and estimation performance of the proposed method can be significantly improved, since the steepest descent step and Barzilai-Borwein step are alternately used as the search step in the unconstrained convex optimization. The proposed method can obtain satisfactory performance especially in these scenarios with low signal to noise ratio (SNR), small number of snapshots, or coherent sources. Simulation results show the superior performance of the proposed method as compared with existing methods.

1. Introduction

Direction of arrival (DOA) estimation of multiple narrowband sources is an important research topic in array signal processing. It has been extensively studied in acoustic source localization, radar, and medical imaging [1–3]. Many effective DOA estimation algorithms have been proposed and developed, which mainly include beamforming algorithms such as MVDR [4] and subspace-based algorithms such as MUSIC [5]. To obtain preferable estimation performance, the Nyquist sampling theorem must be used in these conventional methods of data acquisition. However, high-speed sampling rate can impose so enormous pressure on capturing and storing data that requirements on both hardware and software are increased. Moreover, these methods suffer from serious performance degradation in these scenarios with low signal to noise ratio (SNR), small number of snapshots, or coherent sources.

Recently, many applications involving compressed sensing (CS) [6–8], especially DOA estimation, have been attracting tremendous research interest in the signal processing. CS is an emerging area, and it can not only capture and store compressed or sparse sources simultaneously at a rate much lower than the Nyquist sampling rate but also reconstruct original sources using nonadaptive linear projection measurements onto a suitable measurement matrix, which satisfies the restricted isometry property (RIP) [9–11]. The sparse reconstruction aims to find the support which is shared by the unknown sparse vectors from multiple measurement vectors (MMV). The support denotes the indices of the nonzero elements in the unknown sparse vectors.

CS has been widely applied to DOA estimation since sources are sparse in the spatial domain which results from the fact that there are much fewer true sources directions than all potential directions. Stoica et al. [12] proposed a sparse iterative covariance-based estimation method (SPICE) for array processing which is semiparametric estimation method and can avoid parameter selection. Hyder and Mahata [13] proposed an alternative strategy called joint l_0 approximation (JLZA-DOA) algorithm based on spatial sparsity, which can resolve closely spaced and high correlated sources even if the number of sources is unknown. Figueiredo et al. [14] proposed gradient projection algorithm to solve the bound-constrained quadratic programming formulation. Although it can be simply implemented, the regularization parameter is selected with difficulty and it is only suitable for the single measurement vector (SMV) model, which limits its practical engineering application.

In this paper, we propose a novel multiple sparse measurement gradient reconstruction method called MSMGR for DOA estimation in CS. The method is derived through transforming quadratically constrained linear programming (QCLP) into unconstrained convex optimization to overcome the drawback that l_1-norm is nondifferentiable when minimizing the l_1-norm for the sparse reconstruction. The search steepest descent step [15] and Barzilai-Borwein step [16] are alternately used as the search step to improve the convergence rate and estimation performance significantly. Furthermore, the singular value decomposition (SVD) is incorporated into the proposed method to reduce the computational complexity and the sensitivity to the noise. The proposed method is suitable for both SMV and MMV, and it has higher estimation accuracy and resolution than existing methods especially in these scenarios with low SNR, small number of snapshots, or coherent sources. Simulation results show the superior performance of the proposed method as compared with existing methods.

2. Problem Formulation

Consider K narrowband far-field sources, $\widetilde{s}_k(t)$, $k = 1, 2, \ldots, K$, impinging on the sensor array consisting of N omnidirectional sensors from different directions, $\widetilde{\theta}_i$, $i = 1, 2, \ldots, K$. The array observation model at tth snapshot can be formulated as

$$\mathbf{x}(t) = \sum_{k=1}^{K} \mathbf{a}\left(\widetilde{\theta}_k\right) \widetilde{s}_k(t) + \mathbf{n}(t), \tag{1}$$

where $\mathbf{n}(t) \in \mathbb{C}^{N\times 1}$ is a complex Gaussian white noise vector with zero mean and covariance matrix $\sigma^2\mathbf{I}$ and $\mathbf{a}(\theta_k)$ is the $N \times 1$ steering vector of the source from the direction θ_k. Although the DOA estimation based on the single snapshot, which is a typical SMV model, has its value, the number of snapshots is larger than one in the most practical applications. Correspondingly, the multiple snapshots model is a typical MMV model.

In order to cast the DOA estimation as a sparse reconstruction, let $\{\theta_k\}_{k=1}^{P}$ denote a fine enough grid which covers the entire spatial domain where there are P ($P \gg K$) potential directions of the sources so that the true directions $\{\widetilde{\theta}_k\}_{k=1}^{K}$ are aligned or are close to the grids. It means that there exists $\theta_{k_1}, \theta_{k_2}, \ldots, \theta_{k_K}$ being equal to $\widetilde{\theta}_1, \widetilde{\theta}_2, \ldots, \widetilde{\theta}_K$, respectively. Thus, we have

$$s_k = \begin{cases} \widetilde{s}_l & k = k_l\ (l = 1, 2, \ldots, K) \\ 0 & \text{elsewhere.} \end{cases} \tag{2}$$

The multiple snapshots model can be written as the following sparse form:

$$\mathbf{x}(t) = \sum_{k=1}^{P} \mathbf{a}\left(\theta_k\right) s_k(t) + \mathbf{n}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) \qquad t = 1, 2, \ldots, L, \tag{3}$$

where L is the number of snapshots and $\mathbf{A} = \left[\mathbf{a}(\theta_1)\ \mathbf{a}(\theta_2)\ \cdots\ \mathbf{a}(\theta_P)\right]$ is the $N \times P$ array manifold matrix corresponding to all the potential directions which is also defined as an overcomplete dictionary in CS. $\mathbf{s}(t) = \left[s_1(t)\ s_2(t)\ \cdots\ s_P(t)\right]^T$ is the $P \times 1$ sparse vector with K nonzero elements at positions corresponding to the true directions and zero elements at the remaining $P - K$ positions, where $[\cdot]^T$ denotes the transpose operation. Hence, the matrix $\mathbf{S} = \left[\mathbf{s}(1)\ \mathbf{s}(2)\ \cdots\ \mathbf{s}(L)\right] \in \mathbb{C}^{P\times L}$ has K nonzero rows, that is, row K-sparse, since $\{\mathbf{s}(t)\}_{t=1}^{L}$ share the common support. Obviously, the DOA estimation problem of multiple snapshots is that of identifying the row support of the unknown matrix $\mathbf{S}$ from the matrix $\mathbf{Y} \in \mathbb{C}^{M\times L}$ which is given by

$$\mathbf{Y} = \left[\mathbf{y}(1)\ \mathbf{y}(2)\ \cdots\ \mathbf{y}(L)\right] = \mathbf{\Phi}\mathbf{A}\mathbf{S} + \mathbf{\Phi}\mathbf{N} = \mathbf{\Theta}\mathbf{S} + \mathbf{\Phi}\mathbf{N} \tag{4}$$

with sensing matrix $\mathbf{\Theta}$, noise matrix $\mathbf{N}$, and a common measurement matrix $\mathbf{\Phi}$ of the size $M \times N$ with $M < N$ where M is the number of nonadaptive linear projection measurements.

It is well known that sparse sources can be reconstructed by solving the l_0-norm minimization problem. However, the optimization problem is nonconvex and the optimization method is both numerically unstable and computationally unaccepted [17]. Then, this problem is transformed into the l_1-norm minimization problem [18] so that we can accurately reconstruct the matrix $\mathbf{S}$ by solving the following QCLP problem:

$$\min_{\mathbf{s}} \left\|\mathbf{s}^{(2)}\right\|_1 \qquad \text{s.t.}\ \|\mathbf{Y} - \mathbf{\Theta}\mathbf{S}\|_2^2 \le \sigma^2, \tag{5}$$

where $\mathbf{s}^{(2)}$ is the $P \times 1$ unknown sparse vector and $\|\cdot\|_2$ represents the Frobenius norm of matrices or the Euclidean norm of vectors. The ith entry of $\mathbf{s}^{(2)}$ is equal to Euclidean norm of the ith row of $\mathbf{S}$; that is, $\mathbf{s}_i^{(2)} = \|\mathbf{S}(i, :)\|_2$.

3. DOA Estimation

The SVD is employed on the matrix $\mathbf{Y}$ to reduce the computational complexity and the sensitivity to the noise. Hence, we have

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H = \left[\mathbf{U}_S\ \mathbf{U}_N\right]\mathbf{\Sigma}\mathbf{V}^H, \tag{6}$$

where $\mathbf{V}$ is the orthogonal matrix and $[\cdot]^H$ denotes the conjugate transpose operation. $\mathbf{U}_S$ and $\mathbf{U}_N$ denote the signal subspace and noise subspace, respectively. The M eigenvalues of the matrix $\mathbf{U}$ are arranged from the largest to the smallest; that is, $\sigma_1 \ge \sigma_2 \ge \cdots \sigma_K \gg \sigma_{K+1} \ge \cdots \ge \sigma_M$, where K large eigenvalues are dominant. $\mathbf{U}_S$ and $\mathbf{U}_N$, respectively, consist of left singular eigenvectors corresponding to K big eigenvalues and $M - K$ small eigenvalues. Denote $\mathbf{Y}_S = \mathbf{U}_S = \mathbf{U\Sigma B} = \mathbf{Y\Sigma B} \in \mathbb{C}^{M\times K}$, $\mathbf{S}_S = \mathbf{S\Sigma B}$, and $\mathbf{N}_S = \mathbf{N\Sigma B}$ where $\mathbf{B} = \left[\mathbf{I}_K\ \mathbf{O}_{K\times(L-K)}\right]^H$, $\mathbf{I}_K$ is an identity matrix of the size $K \times K$, and $\mathbf{O}_{K\times(L-K)}$ is a zero matrix of the size $K \times (L - K)$ so that we have

$$\mathbf{Y}_S = \mathbf{\Theta}\mathbf{S}_S + \mathbf{\Phi}\mathbf{N}_S. \tag{7}$$

As one may note, the dimension of the matrix $\mathbf{Y}$ is reduced from $M \times L$ to $M \times K$ which can significantly reduce the computational complexity and the sensitivity to the noise especially in the scenario with large number of snapshots. The essence of the dimension reduction is to keep the signal subspace and discard the noise subspace. By using the SVD decomposition, (5) can be rewritten as the following form:

$$\min_{\mathbf{s}} \left\| \mathbf{s}_S^{(2)} \right\|_1 \qquad \text{s.t.} \left\| \mathbf{Y}_S - \boldsymbol{\Theta} \mathbf{S}_S \right\|_2^2 \le \sigma^2, \tag{8}$$

where $\mathbf{s}_S^{(2)}$ is also a sparse vector and shares the same support with $\mathbf{s}^{(2)}$. To overcome the drawback that l_1-norm is nondifferentiable by minimizing the l_1-norm for solving the sparse sources, (8) is transformed into the unconstrained convex optimization by using Lagrange multiplication [19, 20]:

$$\hat{\mathbf{s}}_S^{(2)} = \arg \min \frac{1}{2} \left\| \mathbf{Y}_S - \boldsymbol{\Theta} \mathbf{S}_S \right\|_2^2 + \tau \left\| \mathbf{s}_S^{(2)} \right\|_1, \tag{9}$$

where τ is nonnegative and it is called regularized factor that can serve as a tradeoff between the ability of suppressing noise and source sparsity. Note that the search path, which is obtained by projecting the negative gradient of the objective function in (9) onto the feasible set, cannot perform a backtracking line search well. Therefore, we adopt the m_1-norm of matrices to change the search direction which results in

$$\hat{\mathbf{S}}_S = \arg \min \frac{1}{2} \left\| \mathbf{Y}_S - \boldsymbol{\Theta} \mathbf{S}_S \right\|_2^2 + \tau \left\| \mathbf{S}_S \right\|_{m_1}, \tag{10}$$

where $\| \cdot \|_{m_1}$ denotes the m_1-norm of matrices. A detailed derivation process of using MSMGR for DOA estimation is shown as follows. Assume that $\mathbf{r}_{\Gamma^{k-1}}$ denotes the residual of the kth iteration, Γ^k denotes the support of the kth iteration, $\boldsymbol{\Theta}_{\Gamma^k}$ denotes the submatrix of $\boldsymbol{\Theta}$ with columns indexed by Γ^k, and $\hat{\mathbf{S}}_{\Gamma^k}$ denotes the reconstructed source after the kth iteration. Therefore, the objective function of the kth iteration can be written as

$$f\left(\hat{\mathbf{S}}_{\Gamma^k}\right) = \frac{1}{2} \left\| \mathbf{Y}_S - \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} \right\|_2^2 + \tau \left\| \hat{\mathbf{S}}_{\Gamma^k} \right\|_{m_1}. \tag{11}$$

The purpose of the current iteration is to find the sparse reconstructed source $\hat{\mathbf{S}}_{\Gamma^k}$ which can minimize the objective function $f(\hat{\mathbf{S}}_{\Gamma^k})$; that is, the residual is minimum after the current iteration. The expansion of (11) can be expressed as

$$\begin{aligned} f\left(\hat{\mathbf{S}}_{\Gamma^k}\right) = \frac{1}{2} \operatorname{tr} \Big[& \mathbf{Y}_S{}^H \mathbf{Y}_S - \mathbf{Y}_S{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} - \hat{\mathbf{S}}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k}{}^H \mathbf{Y}_S \\ & + \hat{\mathbf{S}}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} \Big] + \tau \sum_{i=1}^{k} \sum_{l=1}^{K} |s(i,l)|, \end{aligned} \tag{12}$$

where $s(i, l)$ denotes the element of the ith row and lth column of the matrix $\hat{\mathbf{S}}_{\Gamma^k}$. With further derivation, the minimum of the objective function (12) is equal to the maximum of (13) which can be given as the following form:

$$\begin{aligned} f\left(\hat{\mathbf{S}}_{\Gamma^k}\right) = \frac{1}{2} \operatorname{tr} \Big[& \mathbf{Y}_S{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} + \hat{\mathbf{S}}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k}{}^H \mathbf{Y}_S \\ & - \hat{\mathbf{S}}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} \Big] - \tau \sum_{i=1}^{k} \sum_{l=1}^{K} |s(i,l)|. \end{aligned} \tag{13}$$

Based on the properties of matrix trace, (13) can be further simplified to

$$\operatorname{tr} \left[\mathbf{Y}_S{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{Z}}_{\Gamma^k} - \frac{1}{2} \hat{\mathbf{S}}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} \right] - \tau \sum_{i=1}^{k} \sum_{l=1}^{K} |s(i,l)|. \tag{14}$$

Then, the negative gradient is obtained by the partial derivative of (14) with respect to $\hat{\mathbf{S}}_{\Gamma^k}$ which is given by

$$\begin{aligned} \mathbf{d}_k &= \boldsymbol{\Theta}_{\Gamma^k}{}^H \mathbf{Y}_S - \boldsymbol{\Theta}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} - \tau \frac{d \sum_{i=1}^{k} \sum_{l=1}^{K} |s(i,l)|}{d \hat{\mathbf{S}}_{\Gamma^k}} \\ &= \boldsymbol{\Theta}_{\Gamma^k}{}^H \mathbf{Y}_S - \boldsymbol{\Theta}_{\Gamma^k}{}^H \boldsymbol{\Theta}_{\Gamma^k} \hat{\mathbf{S}}_{\Gamma^k} - \tau \mathbf{D}, \end{aligned} \tag{15}$$

where $\mathbf{D}$ is referred to as polarity matrix that can judge the polarity of nonzero elements:

$$d(i,l) = \begin{cases} 1 + j & \operatorname{Re}\{s(i,l)\} > 0 \ \ \operatorname{Im}\{s(i,l)\} > 0 \\ 1 - j & \operatorname{Re}\{s(i,l)\} > 0 \ \ \operatorname{Im}\{s(i,l)\} < 0 \\ -1 + j & \operatorname{Re}\{s(i,l)\} < 0 \ \ \operatorname{Im}\{s(i,l)\} > 0 \\ -1 - j & \operatorname{Re}\{s(i,l)\} < 0 \ \ \operatorname{Im}\{s(i,l)\} < 0, \end{cases} \tag{16}$$

where $d(i, l)$ denotes the element of the ith row and lth column of the matrix $\mathbf{D}$. Since the conventional search step is too small based on the orthogonality, the steepest descent step and Barzilai-Borwein step are alternately exploited as the search step μ_k in order to improve the convergence rate and estimation performance. Then, we have

$$\mu_k = \begin{cases} \mu_{\text{SD}} & k \text{ is even} \\ \mu_{\text{BB}} & k \text{ is odd}, \end{cases} \tag{17}$$

where μ_{SD} and μ_{BB} are the steepest descent step and Barzilai-Borwein step, respectively. The specific steps of MSMGR are given as follows.

Initialization. Set the number of iterations $k = 1$, $r_{\Gamma^0} = \mathbf{Y}_S$, $\Gamma^0 = \varnothing$, $\boldsymbol{\Theta}_{\Gamma^0} = \varnothing$, and $\hat{\mathbf{S}}_{\Gamma^0} = \varnothing$.

Step 1. Calculate the inner product of the residual of the kth iteration and sensing matrix. Then, update the index p:

$$p = \arg \max_{p} \left\| \left\langle \mathbf{r}^{k-1}, \boldsymbol{\Theta}(:, p) \right\rangle \right\|_2 \quad p = 1, 2, 3, \ldots, P. \tag{18}$$

Step 2. Update the support $\Gamma^k = \Gamma^{k-1} \cup \{p\}$ and the corresponding submatrix $\boldsymbol{\Theta}_{\Gamma^k} = \boldsymbol{\Theta}_{\Gamma^{k-1}} \cup \{\boldsymbol{\Theta}(:, p)\}$.

Step 3. Calculate the negative gradient $\mathbf{d}_k$ and the search step μ_k in terms of (15) and (17), respectively.

Step 4. Update the constructed source $\widehat{\mathbf{S}}_{\Gamma^k} = \widehat{\mathbf{S}}_{\Gamma^{k-1}} + \mathbf{d}_k \cdot \mu_k$ in terms of the negative gradient and the search step.

Step 5. Update the polarity matrix $\mathbf{D} = [\text{sign}(\mathbf{d}_k \cdot \mu_k); \mathbf{O}_{1\times K}]$, where $\mathbf{O}_{1\times K}$ is the zero matrix of the size $1 \times K$.

Step 6. Update the residual $\mathbf{r}^k = \mathbf{r}^{k-1} - \mathbf{\Theta}_{\Gamma^k} \cdot \mathbf{d}_k \cdot \mu_k$. If the residual satisfies the stopping criterion, stop the iteration; otherwise, set $k = k + 1$ and return to Step 1.

The core of the new method is that of updating the polarity matrix by zero-padding process in Step 5 since the dimensions of the support and corresponding submatrix are both expanded in Step 2. Moreover, zero-padding process can guarantee the precision of the DOA estimation. The spectrum of the proposed method is obtained by estimating constructed source power from all potential directions. Like other spectral-based methods, the true directions are estimated by the locations of the highest peaks of the spectrum.

4. Simulation Results

In this section, the superior performance of the proposed method is shown as compared with existing JLZA-DOA and MUSIC methods by several numerical simulations. Consider the spatial sources impinging on the uniform linear array (ULA) with interspacing $\lambda/2$, where λ denotes the wavelength of source. In the ULA case, the steering vector corresponding to the source from the direction θ_k is given by

$$\mathbf{a}(\theta_k) = \left[1 \quad e^{-j\pi \sin(\theta_k)} \quad \cdots \quad e^{-j(N-1)\pi \sin(\theta_k)}\right]^T, \tag{19}$$

where the number of array elements is set to $N = 16$. In the simulation, the regularized factor can be chosen as suggested in [21]

$$\tau = 0.1 \left\| \mathbf{\Theta}^H Y_S \right\|_{m_\infty}. \tag{20}$$

Following [22], it is easy to see that the unique minimum of (10) is the zero matrix for $\tau > \|\mathbf{\Theta}^H Y_S\|_{m_\infty}$. In the simulation, the average root mean square error (RMSE) of the DOA estimation is defined as the significant performance index:

$$\text{RMSE} = \left[\sum_{j=1}^{J} \sum_{k=1}^{K} \frac{\left(\widehat{\theta}_{kj} - \theta_k\right)^2}{JK} \right]^{1/2}, \tag{21}$$

where J is the number of independent Monte Carlo runs and $\widehat{\theta}_{kj}$ is the estimate of θ_k in the jth run. The resolution of the grid is closely related to the precision of the DOA estimation. A coarse grid leads to poor precision, but a too fine grid increases computational complexity. Therefore, an adaptive grid refinement method is used for the tradeoff between precision and computational complexity. In the simulation, we set a coarse grid with 1° step in the range of −90° to 90° and make a local fine grid in the vicinity of the estimated angle.

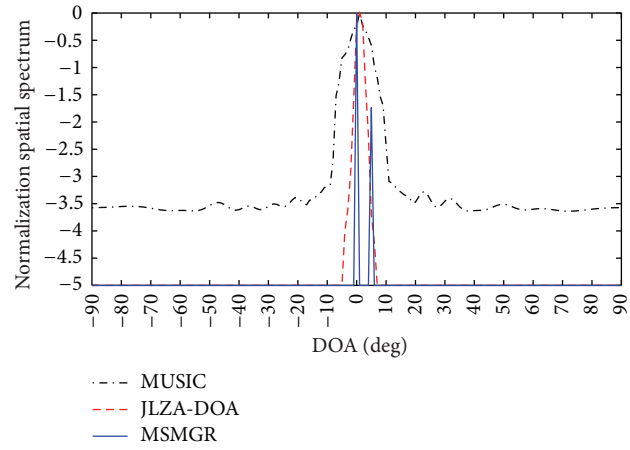


Figure 1: Spatial spectra of MUSIC, JLZA- DOA, and MSMGR for uncorrelated sources.

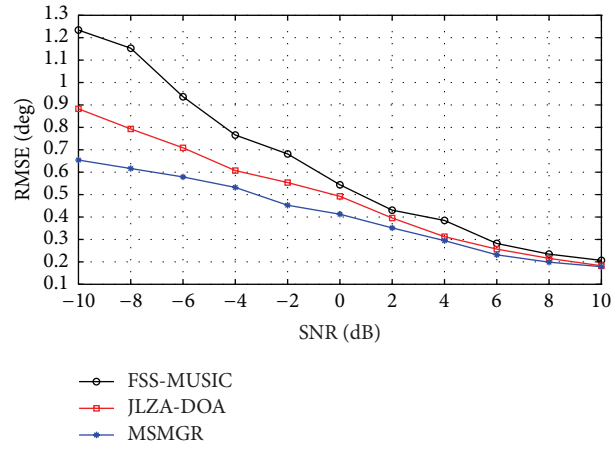


Figure 2: RMSE of the DOA estimation versus SNR for 50 snapshots.

In the first simulation, we show the spatial spectra of three methods in the scenario with low SNR, small number of snapshots, and two uncorrelated sources impinging from $\theta_1 = 0°$ and $\theta_2 = 2°$, respectively. The spatial spectra are shown in Figure 1 with SNR = 3 dB and 50 snapshots. The following conclusion can be acquired from Figure 1 that since the spatial spectra obtained by MUSIC and JLZA-DOA have only one peak, MUSIC and JLZA-DOA cannot identify the closely spaced sources accurately. However, the proposed method MSMGR has a nearly ideal spectrum and a precise estimation for the closely spaced sources. Therefore, MSMGR outperforms JLZA-DOA and MUSIC in terms of the spatial spectrum.

The RMSEs of three methods are analyzed under different conditions in the second simulation. Consider three sources impinging from $\theta_1 = 0°$ $\theta_2 = 49.5°$ and $\theta_3 = 57.3°$, respectively, where the latter two closely spaced sources are correlated and the first source is uncorrelated to them. The forward spatial smoothing method is exploited on the MUSIC called FSS-MUSIC to resolve the correlated sources. Figure 2 shows the RMSE as a function of SNR of three

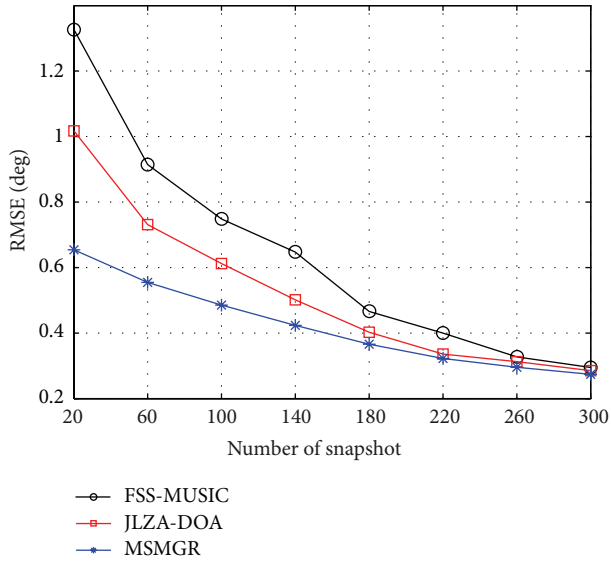


Figure 3: RMSE of the DOA estimation versus number of snapshots for −6 dB SNR.

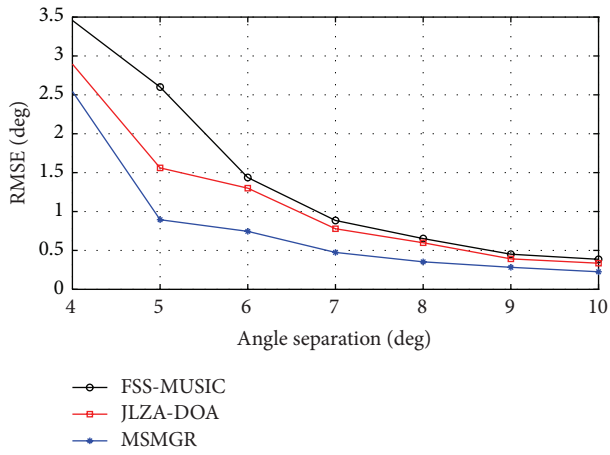


Figure 4: RMSE of the DOA estimation versus angle separation.

methods in 100 Monte Carlo runs for the fixed number of 50 snapshots whereas the RMSE versus the number of snapshots is shown in Figure 3 for the fixed SNR −6 dB in 100 Monte Carlo runs. The conclusions can be drawn from Figures 2 and 3 that the RMSE of MSMGR is smaller than those of other two methods and MSMGR has better estimation performance than the other two methods, especially in the scenarios with low SNR or small number of snapshots. The reason is that MSMGR can overcome the nondifferentiable drawback and exploit the alternate search step to improve the convergence rate and estimation performance. Moreover, the RMSE of MSMGR is close to those of the other two methods with the increase of SNR and the number of snapshots.

In Figure 4, the relation between the RMSE and angle separation of correlated sources is shown, which can illustrate the resolving capability. Let two correlated sources at directions 20° and $20^\circ + \Delta\theta$, where the step of $\Delta\theta$ is 1°, be impinged on the ULA. It can be seen from Figure 4 that MSMGR suffers from serious performance degradation if the angle separation is 4°. However, MSMGR can still provide the most precise estimation as long as the angle separation is no less than 5°. Simulation results show that MSMGR has higher resolution than the other two methods.

Table 1: Computation time (sec) of methods.

Estimation methods	Number of snapshots			
	50	75	100	125
MSMGR	0.685	0.724	0.741	0.779
JLZA-DOA	0.627	0.623	0.645	0.648
FSS-MUSIC	0.423	0.431	0.446	0.451
MSMGR-nosvd	1.521	1.566	1.618	1.654

Finally, we compare the computation time of different methods versus number of snapshots in Table 1. Two correlated sources located at 20° and 27° impinge on the ULA. MSMGR-nosvd denotes that the SVD is not adopted in the process of employing MSMGR for DOA estimation.

It is easy to see from Table 1 that the computation time of MSMGR-nosvd is the longest and the computation time of MSMGR is longer than that of JLZA-DOA and FSS-MUSIC, but it is important to note that the performance of MSMGR is much better than that of these two methods. Moreover, it is proved that the SVD can significantly reduce the computation time.

5. Conclusion

In this paper, a novel MSMGR method for DOA estimation is proposed in CS. The proposed method is obtained by transforming QCLP into unconstrained convex optimization to overcome the drawback that l_1-norm is nondifferentiable when sparse sources are reconstructed by minimizing the l_1-norm. An alternate search step is used to improve the convergence rate and estimation performance. The SVD is used to reduce the computational complexity and the sensitivity to the noise. Simulation results show that MSMGR outperforms JLZA-DOA and MUSIC in terms of the spatial spectrum and has more precise estimation as well as higher resolution; in particular when SNR is low, the number of snapshots is small and sources are coherent.

Conflict of Interests

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

Acknowledgment

This work was supported by the Fundamental Research Funds for the Center University of China under Grant no. HEUCF130804.

References

[1] V. V. Reddy, M. Mubeen, and B. P. Ng, "Reduced-Complexity Super-Resolution DOA estimation with unknown number of sources," *IEEE Signal Processing Letters*, vol. 22, no. 6, pp. 772–776, 2015.

[2] Z. Tan and A. Nehorai, "Sparse direction of arrival estimation using co-prime arrays with off-grid targets," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 26–29, 2014.

[3] P. Falcone, F. Colone, A. Macera, and P. Lombardo, "Two-dimensional location of moving targets within local areas using WiFi-based multistatic passive radar," *IET Radar, Sonar & Navigation*, vol. 8, no. 2, pp. 123–131, 2014.

[4] J. P. Burg, "Maximum entropy spectral analysis," in *Proceedings of the 37th Meeting of the Society of Exploration Geophsicists*, 1967.

[5] R. O. Schmidt, *A Signal Subspace Approach to Multiple Emitter Location and Spectrum Estimation*, Stanford University, Stanford, Calif, USA, 1981.

[6] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[7] J. D. Blanchard, J. Tanner, and K. Wei, "Conjugate gradient iterative hard thresholding: observed noise stability for compressed sensing," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 528–537, 2015.

[8] K. Sano, R. Matsushita, and T. Tanaka, "To average or not to average: trade-off in compressed sensing with noisy measurements," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '14)*, pp. 1316–1320, Honolulu, Hawaii, USA, June-July 2014.

[9] P. Koiran and A. Zouzias, "Hidden cliques and the certification of the restricted isometry property," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4999–5006, 2014.

[10] A. M. Tillmann and M. E. Pfetsch, "The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1248–1259, 2014.

[11] C. B. Song and S. T. Xia, "Sparse signal recovery by l_p minimization under restricted isometry property," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1154–1158, 2014.

[12] P. Stoica, P. Babu, and J. Li, "SPICE: a sparse covariance-based estimation method for array processing," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 629–638, 2011.

[13] M. M. Hyder and K. Mahata, "Direction-of-arrival estimation using a mixed $l_{2,0}$ norm approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4646–4655, 2010.

[14] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal on Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.

[15] J. Fliege and B. F. Svaiter, "Steepest descent methods for multicriteria optimization," *Mathematical Methods of Operations Research*, vol. 51, no. 3, pp. 479–494, 2000.

[16] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[17] C. L. Bao, H. Ji, Y. H. Quan, and Z. W. Shen, "L_0 norm based dictionary learning by proximal methods with global convergence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 3858–3865, Columbus, Ohio, USA, June 2014.

[18] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for l_1-subspace signal processing," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5046–5058, 2014.

[19] J. Sandoval-Moreno, G. Besancon, and J. J. Martinez, "Lagrange Multipliers based price driven coordination with constraints consideration for multisource power generation systems," in *Proceedings of the European Control Conference (ECC '14)*, pp. 1987–1992, IEEE, Strasbourg, France, June 2014.

[20] Z.-Q. Lü and X. An, "Non-conforming finite element tearing and interconnecting method with one Lagrange multiplier for solving large-scale electromagnetic problems," *IET Microwaves, Antennas & Propagation*, vol. 8, no. 10, pp. 730–735, 2014.

[21] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinvesky, "An interior-point method for large-scale l_1—regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 606–616, 2007.

[22] J. J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1341–1344, 2004.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 810798, 10 pages
http://dx.doi.org/10.1155/2015/810798

Research Article

A Robust Optimization of Capacity Allocation Policies in the Third-Party Warehouse

Xu Xian-hao,[1] Dong Wei-hong,[1] and Peng Hongxia[2]

[1]*School of Management, Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*School of Business, Hubei University, Wuhan 430062, China*

Correspondence should be addressed to Dong Wei-hong; dongweihong@hust.edu.cn

Received 10 October 2014; Accepted 15 January 2015

Academic Editor: John Gunnar Carlsson

We study the capacity allocation policies of a third-party warehouse center, which supplies several different level services on different prices with fixed capacity, on revenue management perspective. For the single period situation, we use three different robust methods, absolute robust, deviation robust, and relative robust method, to maximize the whole revenue. Then we give some numerical examples to verify the practical applicability. For the multiperiod situation, as the demand is uncertain, we propose a stochastic model for the multiperiod revenue management problem of the warehouse. A novel robust optimization technique is applied in this model to maximize the whole revenue. Then we give some numerical examples to verify the practical applicability of our method.

1. Introduction and Literature Review

In today's business world, a large number of companies outscore their warehouse functions to the third-party Warehouse (3PW) company in order to minimize their operation costs and focus on their core competencies. Therefore, warehousing industry becomes a booming business all over the world. According to the survey data from National Bureau of Statistics (NBS) of China, the national warehousing investment in fixed assets amounted to 69.20 billion dollars in 2013, increasing 32.7% over 2012. With the fast development of third-party warehousing industry, the revenue problem has received considerable attentions from both 3PW practitioners and researchers. 3PW company can provide storage services to different customers with fixed storage capacity and then capacity allocation policy plays an important role in revenue management.

The aim of capacity allocation in 3PW is to pursue a better fit between storage capacity allocation and market demand for each level in order to improve the expected revenue. In addition, customer demands for each level are uncertain. In this paper, we focus on the capacity allocation policy of a 3PW company for both single storage period and multiperiod with a revenue management perspective and robust optimization method.

Revenue management (RM) is a useful tool to help companies sell their products or services to right customers at right price and right time and make greatest revenue [1]. The field of revenue management is originated in the airline industry as a way to efficiently allocate fixed capacity to different classes of customers [2]. Talluri and Van Ryzin [3] discussed the network RM problems extensively, and their work was based on the independent demand model. The success of airline revenue management was widely reported, and this stimulated development of revenue management systems for other transportation sectors and in other areas of the services sectors, such as automobile rental [4], broadcasting [5], and hospitality [6]. Chiang et al. [7] provided an overview paper on revenue management. The solution methods that are mentioned in these papers are mainly mathematical programming, dynamic programming, and heuristics such as genetic algorithm. Revenue management is applied into many industries but not warehouse. In this paper, we study the capacity allocation problem with the perspective on revenue management.

Several researchers have worked at third-party warehousing. Gong and de Koster [8] gave a review on stochastic models and analysis on warehouse operations. Analysis of third-party warehousing contracts with commitments was studied without revenue consideration with capacity allocation in Chen et al. [9]. Lin [10] studied capacity allocation policy of third-party warehouse with dynamic optimization in revenue management perspective. Zhang et al. [11] provided the mathematical model of allocating customers to different warehouse spaces with deterministic demand and they solved this problem with a scheduling approach. Gong et al. [12] proposed a facility design method to improve the warehouse revenue with the consideration of stochastic market demand and the customers arrived according to a Poisson process. However, it is difficult to characterize the distribution of the uncertain demand, and robust optimization with uncertain demand is not included in these papers.

Robust optimization is a useful method to solve stochastic programming with unknown probability. Soyster [13] first proposed factor uncertainty in the field of optimization and gave its robust method. Ben-Tal and Nemirovski [14–16] proposed several uncertainties regarding the form and its applications and analyzed the robust methods of the linear programming and general convex programming. El Ghaoui et al. [17, 18] derived a similar conclusion and gave the robust methods to solve optimization problems in practical application. Bertsimas et al. [19, 20] did a further research on the basis of their work; they relaxed the conservation of the robust optimization and introduced the concept of the price of robust. Vairaktarakis [21] presented robust newsboy models with uncertain demand and provided an alternative approach using deterministic optimization models which could be solved by dynamic programming. Their work studied the single period problem, and our paper can solve the capacity allocation problem for both single period and multiple periods.

The rest of this paper is organized as follows. In Section 2, we describe the capacity allocation problem of the 3PW company and the corresponding mathematical model. Then we propose three different robust models of single period problem, the corresponding algorithms, and we give some numerical examples in Section 3. Section 4 presents the robust capacity allocation policies for multiperiod situation and then gives some illustrative examples. Finally, Section 5 concludes the paper with a short summary and future direction.

2. Problem Description

In this paper, we consider such a 3PW company which provides several different levels of warehousing service for customers, with fixed capacity L. The demand in each level is uncertain. The service price or cost for each storage level is different; therefore the unit revenue for each level is quite different. Facing such condition, this company should decide the storage capacity for each level. As the total capacity is fixed, if we increase one level's capacity, the opportunity cost of other levels maybe occur; if not, satisfaction degree of customer in this level may decrease and give orders to other competitors. In order to improve the revenue of this warehouse center and use the storage capacity more appropriately, they should provide an appropriate capacity allocation policy which can meet the market demand more accurately. Therefor, this capacity allocation problem is how this 3PW company will allocate its limited warehousing capacity to each storage service level; then it can maximize the total revenue.

Hypotheses are made as the following:

L: total capacity of the 3PW company;

N: number of the service levels;

L_n: capacity number of nth level, $n = 1, \ldots, N$;

P_n: the unit price of nth level per period;

C_n: the unit cost of nth level per period, and $C_n \leqslant P_n$;

S_n: the unit cost of lost sales of nth level per period;

D_n: the demand of nth level capacity, it is an interval uncertain variable with the probability density function $f(\cdot)$ and cumulative distribution function $F(\cdot)$;

$\pi_n(L_n, D_n)$: the revenue of nth level capacity with capacity L_n and demand D_n.

The TPW is a unit-load warehouse; that is, all goods in this warehouse need to occupy the same storage space (one pallet); split of the pallet does not exist.

The profit function for the ith item is given by

$$\pi_n(L_n, D_n) = \begin{cases} P_n D_n - C_n L_n, & L_n \geqslant D_n \\ P_n L_n - \left(C_n L_n + S_n\left(D_n - L_n\right)\right), & L_n < D_n. \end{cases} \tag{1}$$

In a similar way as stochastic knapsack method, dynamic warehousing capacity allocation model is obtained as follows:

$$\begin{aligned} \max \quad & E\left(\pi_n\left(L_n, D_n\right)\right) \\ & = \int_0^{L_n} \left(P_n u_n - C_n L_n\right) f\left(u_n\right) du_n \\ & \quad + \int_{L_n}^{\infty} \left(P_n L_n - \left(C_n L_n + S_n\left(u_n - L_n\right)\right)\right) f\left(u_n\right) du_n \\ \text{s.t.} \quad & \sum_{n=1}^{N} L_n \leqslant L \\ & L_n \geqslant 0, \quad n = 1, \ldots, N. \end{aligned} \tag{2}$$

The first constraint is the total capacity constraint, and the second one implies capacity of each level cannot be negative.

Theorem 1. *There exists the optimum solution in formula (2) without constraint.*

Proof. Analyzing formula (2), we can get the first-order derivative as follows:

$$\begin{aligned}\frac{\partial E}{\partial L_n} &= \frac{\partial}{\partial L_n}\left(\int_0^{L_n}\left(P_n u_n - C_n L_n\right) f\left(u_n\right) du_n + \int_{L_n}^{\infty}\left(P_n L_n - \left(C_n L_n + S_n\left(u_n - L_n\right)\right)\right) f\left(u_n\right) du_n\right)\\ &= \frac{\partial}{\partial L_n}\left(P_n \int_0^{L_n} u_n f\left(u_n\right) du_n - S_n \int_{L_n}^{\infty} u_n f\left(u_n\right) du_n + \left(P_n + S_n\right) L_n\left(1 - F\left(L_n\right)\right) - C_n L_n\right)\\ &= P_n L_n f\left(L_n\right) + S_n L_n f\left(L_n\right) + \left(P_n + S_n\right)\left(1 - F\left(L_n\right)\right) - \left(P_n + S_n\right) L_n f\left(L_n\right) - C_n\\ &= \left(P_n + S_n - C_n\right) - \left(P_n + S_n\right) F\left(L_n\right).\end{aligned} \tag{3}$$

And the second-order derivative is

$$\frac{\partial^2 E}{\partial {L_n}^2} = -\left(P_n + S_n\right) f\left(L_n\right) \leqslant 0. \tag{4}$$

Now we know that the expected revenue function is a concave function about variable L_n, so there exists the optimum solution if there is no constraint condition, and it should satisfy the following condition:

$$F\left({L_n}^*\right) = \frac{P_n + S_n - C_n}{P_n + S_n}. \tag{5}$$

However, in practice we can hardly know the cumulative distribution function $F(\cdot)$ of the demand variable D_n. On the contrary, the maximum and minimum value of D_n can easily be got. In the rest of this paper, we discuss the interval demand condition with $D_n \in [\underline{D_n}, \overline{D_n}]$. □

3. Robust Optimization for Single Period

There exist many methods to describe the uncertainty in management optimization problems. One of the most classic versions is the assumption that the probability distribution of the random variable is known. However, it is always not realistic in the actual problem. Robust optimization is a useful method to solve stochastic programming with unknown probability.

According to Vairaktarakis [21], there are three different types of robust methods: absolute robust, deviation robust, and relative robust method. Applying these three robust methods in warehousing capacity allocation, we can get the following three models.

3.1. Absolute Robust Model. In case that the demand realizations for item n take values from the interval $D_n \in [\underline{D_n}, \overline{D_n}]$, our absolute robust formulation with a budget constraint becomes

$$\max_{L_n} \min_{D_n \in [\underline{D_n}, \overline{D_n}]} \pi_n\left(L_n, D_n\right). \tag{6}$$

Analyzing the objective function, we can get

$$\begin{aligned}&\min_{D_n \in [\underline{D_n}, \overline{D_n}]} \pi_n\left(L_n, D_n\right)\\ &= \min_{D_n \in [\underline{D_n}, \overline{D_n}]} \begin{cases} P_n D_n - C_n L_n, & L_n \geqslant L_n^A \geqslant \underline{D_n} \\ P_n L_n - \left(C_n L_n + S_n\left(D_n - L_n\right)\right), & L_n \leqslant L_n^A \leqslant \overline{D_n} \end{cases}\\ &= \begin{cases} P_n \underline{D_n} - C_n L_n, & L_n \geqslant L_n^A \geqslant \underline{D_n} \\ \left(P_n + S_n - C_n\right) L_n - S_n \overline{D_n}, & L_n \leqslant L_n^A \leqslant \overline{D_n}. \end{cases}\end{aligned} \tag{7}$$

Thus, the absolute robust allocation L_n^A should satisfy the following equation:

$$\pi_n\left(L_n^A, \underline{D_n}\right) = \pi_n\left(L_n^A, \overline{D_n}\right). \tag{8}$$

That implies

$$L_n^A = \frac{P_n \underline{D_n} + S_n \overline{D_n}}{P_n + S_n}. \tag{9}$$

Now, we can get the absolute robust allocation model with uncertain interval demand as follows:

$$\begin{aligned}\max_{L_1,\ldots,L_N} \min_{D_n \in [\underline{D_n}, \overline{D_n}]} \quad & \sum_{n=1}^{N} \pi_n\left(L_n, D_n\right)\\ \text{s.t.} \quad & \sum_{n=1}^{N} L_n \leqslant L.\end{aligned} \tag{10}$$

The following observations can be made for model AR.

Theorem 2. *There exists an optimal solution* $\{L_1^{A*}, \ldots, L_N^{A*}\}$ *for this AR model, and* $L_n^{A*} \in [\underline{D_n}, L_n^A]$, $n = 1, \ldots, N$.

With this theorem, we can get the equivalent form of formula (10):

$$\max_{L_1,\ldots,L_n} \sum_{n=1}^{N} \left(\left(P_n + S_n - C_n\right) L_n - S_n \overline{D_n}\right). \tag{11}$$

The optimal solution of (11) maximizes the quantity $\sum_{n=1}^{N}(P_n + S_n - C_n) L_n$, and therefore AR model can be reduced to a continuous knapsack problem. The corresponding algorithm will be introduced in the next section.

3.2. Deviation Robust Model. The deviation robust-order quantity is the solution of

$$\min_{L_n} \max_{D_n \in [\underline{D_n}, \overline{D_n}]} \left(\pi_n(D_n, D_n) - \pi_n(L_n, D_n)\right). \tag{12}$$

This formulation provides a solution that minimizes over all choices of order quantities the maximum profit loss due to demand uncertainty. This is a mini-max regret approach where the regret is captured by the difference $\pi_n(D_n, D_n) - \pi_n(L_n, D_n)$. The profit could be realized if there is no demand uncertainty in which case we would order $L_n = D_n$.

That equals

$$\begin{aligned}
&\max_{D_n \in [\underline{D_n}, \overline{D_n}]} \left(\pi_n(D_n, D_n) - \pi_n(L_n, D_n)\right) \\
&= \max_{D_n \in [\underline{D_n}, \overline{D_n}]} \begin{cases} (P_n D_n - C_n D_n) - (P_n D_n - C_n L_n), \\ \qquad\qquad L_n \geqslant L_n^D \geqslant \underline{D_n} \\ ((P_n + S_n - C_n) D_n - S_n D_n) \\ \quad - ((P_n + S_n - C_n) L_n - S_n D_n), \\ \qquad\qquad L_n \leqslant L_n^D \leqslant \overline{D_n} \end{cases} \\
&= \max_{D_n \in [\underline{D_n}, \overline{D_n}]} \begin{cases} C_n L_n - C_n D_n, & L_n \geqslant L_n^D \geqslant \underline{D_n} \\ (P_n + S_n - C_n) D_n \\ \quad - (P_n + S_n - C_n) L_n, & L_n \leqslant L_n^D \leqslant \overline{D_n} \end{cases} \\
&= \begin{cases} C_n L_n - C_n \underline{D_n}, & L_n \geqslant L_n^D \geqslant \underline{D_n} \\ (P_n + S_n - C_n) \overline{D_n} \\ \quad - (P_n + S_n - C_n) L_n, & L_n \leqslant L_n^D \leqslant \overline{D_n}. \end{cases}
\end{aligned} \tag{13}$$

Thus, the deviation robust allocation should satisfy the following equation:

$$L_n^D = \frac{(P_n + S_n - C_n)\overline{D_n} + C_n \underline{D_n}}{P_n + S_n}. \tag{14}$$

Now, we can get the absolute robust allocation model with uncertain interval demand as follows:

$$\begin{aligned}
&\min_{L_1, \ldots, L_N} \max_{D_n \in [\underline{D_n}, \overline{D_n}]} \quad \sum_{n=1}^{N} \left(\pi_n(D_n, D_n) - \pi_n(L_n, D_n)\right) \\
&\text{s.t.} \qquad\qquad \sum_{n=1}^{N} L_n \leqslant L.
\end{aligned} \tag{15}$$

Just as AR model, the objective function equals

$$\min_{L_1, \ldots, L_n} \sum_{n=1}^{N} \left((P_n + S_n - C_n)\overline{D_n} - (P_n + S_n - C_n) L_n\right). \tag{16}$$

It makes us maximize $\sum_{n=1}^{N}(P_n + S_n - C_n)L_n$. Therefore DR model reduces to a continuous knapsack problem. The corresponding algorithm will be introduced in the next section.

3.3. Relative Robust Model. The third robust formulation is called relative robustness and the corresponding formulation is given by

$$\min_{L_n} \max_{D_n \in [\underline{D_n}, \overline{D_n}]} \frac{\pi_n(D_n, D_n) - \pi_n(L_n, D_n)}{\pi_n(D_n, D_n)} \tag{17}$$

which minimizes the relative profit loss per unit of profit that could be made if there was no demand uncertainty. Note that the relative profit loss measures the lost profit as a percentage of the profit that could be made if we knew the actual demand.

In the rest of our analysis it will become clear that the three objectives result in very different choices of order quantities. Similar formulations can be written for the case of interval scenarios. The only difference in modeling the continuous case is that there is a constraint $D_n \in [\underline{D_n}, \overline{D_n}]$.

As we have analyzed above, it equals

$$\begin{aligned}
&\max_{D_n \in [\underline{D_n}, \overline{D_n}]} \frac{\pi_n(D_n, D_n) - \pi_n(L_n, D_n)}{\pi_n(D_n, D_n)} \\
&= \max_{D_n \in [\underline{D_n}, \overline{D_n}]} \begin{cases} \dfrac{(P_n D_n - C_n D_n) - (P_n D_n - C_n L_n)}{P_n D_n - C_n D_n}, \\ \qquad\qquad L_n \geqslant L_n^R \geqslant \underline{D_n} \\ (((P_n + S_n - C_n) D_n - S_n D_n) \\ \quad - ((P_n + S_n - C_n) L_n - S_n D_n)) \\ \quad \cdot ((P_n + S_n - C_n) D_n - S_n D_n)^{-1}, \\ \qquad\qquad L_n \leqslant L_n^R \leqslant \overline{D_n} \end{cases} \\
&= \begin{cases} \dfrac{C_n L_n - C_n \underline{D_n}}{P_n \underline{D_n} - C_n \underline{D_n}}, & L_n \geqslant L_n^R \geqslant \underline{D_n} \\ \dfrac{(P_n + S_n - C_n)\overline{D_n} - (P_n + S_n - C_n) L_n}{P_n \overline{D_n} - C_n \overline{D_n}}, & L_n \leqslant L_n^R \leqslant \overline{D_n}. \end{cases}
\end{aligned} \tag{18}$$

The last equation has the optimum solution if and only if $L_n^R = (P_n + S_n)\overline{D_n}\underline{D_n}/((P_n + S_n - C_n)\underline{D_n} + C_n\overline{D_n})$.

Finally, we can get the RR model:

$$\begin{aligned}
&\min_{L_1, \ldots, L_N} \max_{D_n \in [\underline{D_n}, \overline{D_n}]} \frac{\pi_n(D_n, D_n) - \pi_n(L_n, D_n)}{\pi_n(D_n, D_n)} \\
&\text{s.t.} \qquad\qquad \sum_{n=1}^{N} L_n \leqslant L.
\end{aligned} \tag{19}$$

That equals

$$\begin{aligned}
&\min_{L_1, \ldots, L_N} \sum_{n=1}^{N} \frac{(P_n + S_n - C_n)\overline{D_n} - (P_n + S_n - C_n) L_n}{P_n \overline{D_n} - C_n \overline{D_n}} \\
&= \min_{L_1, \ldots, L_N} \sum_{n=1}^{N} \frac{(P_n + S_n - C_n)\overline{D_n}}{P_n \overline{D_n} - C_n \overline{D_n}} - \frac{(P_n + S_n - C_n)}{P_n \overline{D_n} - C_n \overline{D_n}} L_n.
\end{aligned} \tag{20}$$

Table 1: Each cost of warehouse A.

Level	P_n	C_n	S_n	$\underline{D_n}$	$\overline{D_n}$	$P_n + S_n - C_n$	$(P_n + S_n - C_n)/(P_n\overline{D_n} - C_n\overline{D_n})$
1	10	4	4	500	700	10	0.00238
2	8	2	3	600	700	9	0.00214
3	12	5	5	200	300	12	0.00686
4	14	4	3	300	450	13	0.00289

Table 2: The capacity allocation policies of 3 types of robust model.

Level	Prior (A/DR)	Prior (RR)	L_n^A	L_n^{A*}	L_n^D	L_n^{D*}	L_n^R	L_n^{R*}
1	3	3	557	557	643	643	628	628
2	4	4	627	388	682	171	679	207
3	2	1	229	229	271	271	262	262
4	1	2	326	326	415	415	403	403

3.4. Algorithm and Illustrative Examples. In this section, we adapt the continuous knapsack procedure to the three robust formulations.

Algorithm for AR Model

Step 1. Define the weight $w_n^A = P_n + S_n - C_n$, $n = 1, \ldots, N$, and index the items such that $w_1^A \geqslant w_2^A \geqslant \cdots \geqslant w_N^A$.

Step 2. If $L \geqslant \sum_{n=1}^{N} L_n^A$, then $L_n^{A*} = L_n^A$, $n = 1, \ldots, N$.

Otherwise identify the critical item t such that

$$t = \arg\min_i \left\{ \sum_{j=1}^{N} \underline{D_j} + \sum_{j=1}^{i} \left(L_j^A - \underline{D_j} \right) \geqslant L \right\}. \quad (21)$$

Step 3. If $i < t$, $L_i^{A*} = L_i^A$; if $i > t$, $L_i^{A*} = \underline{D_i}$; if $i = t$, $L_i^{A*} = L - \sum_{j \neq t} L_j^{A*}$.

As the algorithm for DR model is the same as the above algorithm, we do not show it again here.

3.5. Algorithm for RR Model

Step 1. Define the weight $w_n^R = (P_n + S_n - C_n)/(P_n\overline{D_n} - C_n\overline{D_n})$, $n = 1, \ldots, N$, and index the items such that $w_1^R \geqslant w_2^R \geqslant \cdots \geqslant w_N^R$.

Step 2. If $L \geqslant \sum_{n=1}^{N} L_n^R$, then $L_n^{R*} = L_n^R$, $n = 1, \ldots, N$.

Otherwise identify the critical item t such that

$$t = \arg\min_i \left\{ \sum_{j=1}^{N} \underline{D_j} + \sum_{j=1}^{i} \left(L_j^R - \underline{D_j} \right) \geqslant L \right\}. \quad (22)$$

Step 3. For $i < t$, $L_i^{R*} = L_i^R$; for $i > t$, $L_i^{R*} = \underline{D_i}$; for $i = t$, $L_i^{R*} = L - \sum_{j \neq t} L_j^{R*}$.

3.6. Illustrative Examples. There is a third-party warehouse company A, the total capacity is 1500, and the service price and cost for each level are shown as Table 1. Assume that, for each level n, we are given a scenario D_n of demand quantities that may be realized. The number of likely minimum and maximum demand of D_n is $\underline{D_n}$ and $\overline{D_n}$. Then we can calculate the weights w_n^A/w_n^D and w_n^R, which are shown in the last two columns.

By Table 1, we can get the prior list of each robust model. Then using the algorithm that we have proposed in last section, we can get the formulation for the three types of robust policies which are shown as Table 2.

Table 2 indicates that level 4 receives top priority by AR/DR because it results in the highest (profit + lost sale) ratio; level 3 receives top priority by RR as it holds the higher (profit + lost sale) ratio and lower $\underline{D}$; level 3 is favored by the RR objective while its priority is lower for the AR and DR objectives. Similar conclusions can be made for the remaining items.

As a result, for any budget level and for every one of the three objectives, our formulations result to order the maximum possible number of units starting with high priority items and continue on with items of lower priority. For example, in the relative robust policy, we should first satisfy the first three highest priority levels: level 4, level 3, and level 1 with the capacity allocations 326, 529, and 557. For level 4, the only left capacity is 388. Similarly, we can get the deviation robust policy and relative robust policy, which are shown as the seventh and ninth column in Table 2.

3.7. Analysis of the Three Robust Policies. In this section, we analyze the effect of these three robust policies on the company revenue. In general, we use three scenarios to simulate the demand market: scenario 1, the lowest situation, the demand of each level is the minimum demand, that is, 500, 600, 200, and 300; scenario 2, the highest situation, the demand of each level is the maximum demand, that is, 700, 700, 300, and 450; scenario 3, the middle situation, the demand of each level is 600, 650, 250, and 375. Using the three robust policies in Table 2, we can get the total revenue of the

Table 3: The revenue of each policy.

Robust policy	Scenario 1	Scenario 2	Scenario 3	Average revenue
AR	8615	8298	9323	8745.3
DR	5752	8866	8594	7737.3
RR	6229	8776	8729	7911.3

third-party warehouse company under each robust policy in Table 3.

From Table 3, we can get that, for scenario 1 (the minimum demand situation), the best policy is absolute robust policy, and the best revenue is 8615, while the worst revenue of the deviation robust policy is 5752. For the maximum demand situation, the best policy is the deviation robust policy, and the best revenue is 8866, while the worst revenue of the absolute robust policy is 8298. For the middle demand situation, the best revenue (9323) comes from the absolute robust policy, while the worst revenue (8594) comes from the deviation robust policy. And we can express Table 3 by Figure 1.

By Figure 1, we can observe that, in both the minimum and middle demand situation, the absolute robust policy is better than the relative robust policy, while the relative robust policy is better than deviation robust policy. However, the opposite result appears in the middle demand situation. From the average revenue, the best policy is absolute robust policy while the worst policy is the deviation robust policy.

The results make sense for the 3PW company holder. On one hand, if the demand market is not so high, that is, the demand for each level is lower, he should choose the absolute robust policy and avoid deviation robust policy. Otherwise, he should choose deviation robust or relative robust policy when the demand for each level is high. On the other hand, from the perspective of average revenue, the absolute robust policy is the best policy for the conservative holders, whose managements are risk aversion.

4. Robust Optimization for Multiple Periods

In this section, we extend this capacity allocation problem into multiperiod condition. The following are the new hypotheses which are used in this section, and the remaining parameters are the same as last section.

D_{ijn}: storage demand which starts from ith period and ends at jth period of nth service level ($0 \le i < j \le T$), and it is an uncertain variable;

r_n: the unit revenue of storage service level n;

L_{ijn}: decision variables, the capacity allocation which starts from ith period and ends at jth period of nth service level;

$\sum_{j=i+1}^{T} L_{ijn}$: the number of products of nth service level which are stored on the ith period;

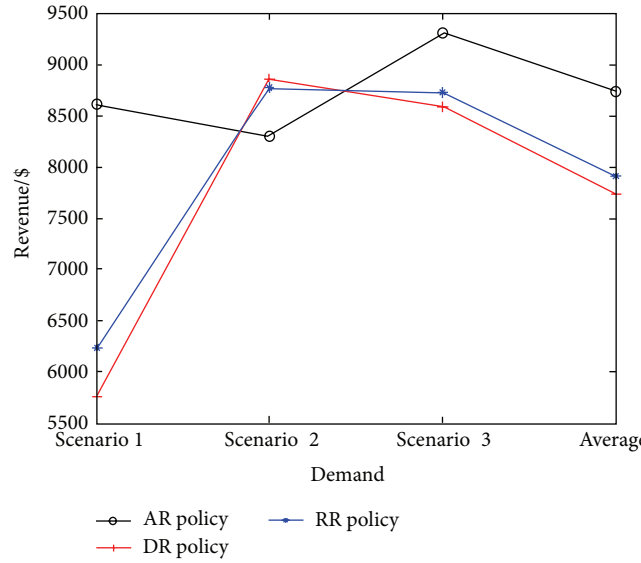


Figure 1: The revenue of each policy under each scenario.

$\sum_{i=0}^{j-1} L_{ijn}$: the number of products of nth service level which are retrieved on the jth period.

We assume there are no goods staying before period 0 and all the goods have to be retrieved on or before the last period. The 3PW is a unit-load warehouse; that is, all goods in this warehouse need to occupy the same storage space (one pallet); split of the pallet does not exist.

We consider a particular period t, $t = 1, 2, \ldots, T-1$; the following equation models the occupation status of the warehouse center on period t:

$$\sum_{i=0}^{t-1}\sum_{j=t}^{T} L_{ijn} + \sum_{j=t+1}^{T} L_{tjn} - \sum_{i=0}^{t-1} L_{itn}. \tag{23}$$

The first part of this equation stands for the number of goods which stay over period t, that is, storage before period t and retrieve after period t; the second part means the number of goods which are stored on period t; the last part means the number of the goods which retrieve on period t.

With the fixed capacity, we have the following constraints for period t:

$$\sum_{i=0}^{t-1}\sum_{j=t}^{T} L_{ijn} + \sum_{j=t+1}^{T} L_{tjn} - \sum_{i=0}^{t-1} L_{itn} \le L_n. \tag{24}$$

Particularly, on the period 0, we have

$$\sum_{j=1}^{T} L_{0jn} \le L_n, \quad n = 1, 2, \ldots, N. \tag{25}$$

Then we can get the stochastic mathematical model as follows:

$$\begin{aligned}
\max \quad & \sum_{n=1}^{N}\sum_{i=0}^{T-1}\sum_{j=i+1}^{T} r_n \cdot (j-i) L_{ijn} \\
\text{s.t.} \quad & \sum_{i=0}^{t-1}\sum_{j=t}^{T} L_{ijn} + \sum_{j=t+1}^{T} L_{tjn} - \sum_{i=0}^{t-1} L_{itn} \le L_n, \\
& t = 1, 2, \ldots, T-1, \quad n = 1, 2, \ldots, N; \\
& \sum_{j=1}^{T} L_{0jn} \le L_n, \quad n = 1, 2, \ldots, N; \\
& \sum_{n=1}^{N} L_n \le L; \\
& L_{ijn} \le D_{ijn}, \quad D_{ijn} \in N \cup \{0\}, \\
& n = 1, 2, \ldots, N, \quad 0 \le i < j \le T.
\end{aligned} \tag{26}$$

The objective function is to maximize the total revenue of all period and all service level. The third constraint condition is the total capacity constraint; it means that in every time period the sum of capacity allocations of all service level can not be larger than the total capacity of the 3PW. The last constraint condition stands for the fact that the capacity allocation variable L_{ijn} must be an integer which is not smaller than 0.

4.1. Robust Optimization Model. The problem looks like a linear integer programming problem. Unfortunately, the parameters D_{ijn} are usually uncertain at the beginning of planning period. Moreover, the revenues may not be fixed, as the decision maker would like to set different pricing, which in turn results in different demands. One may want to solve this by replacing the parameters by their best point estimator, for instance, using expected value $E[D_{ijn}]$ to replace the uncertain parameter of D_{ijn}. One of the most classic versions is the assumption that the probability distribution of the random variable is known. However, it is always not realistic in the actual problem. Robust optimization is a useful method to solve stochastic programming with unknown probability. We believe decision makers would prefer to use proactive tools to obtain their solutions.

According to Mulvey et al. [22], there are two definitions about robustness.

Definition 3 (solution robustness). An optimal solution is solution robust with respect to optimality if it remains "close" to being optimal for any scenario $s \in \theta$.

Definition 4 (model robustness). An optimal solution is model robust with respect to feasibility if it remains "almost" feasible for any scenario $s \in \theta$.

Consider such a stochastic programming:

$$\begin{aligned}
\min \quad & c^T x + d^T y \\
\text{s.t.} \quad & Ax = b \\
& Bx + Cy = 0 \\
& x \ge 0, \quad y \ge 0,
\end{aligned} \tag{27}$$

where the decision variable y contains uncertainty. Then the corresponding robust model can be written as follows:

$$\begin{aligned}
\min \quad & \sigma(x, y_1, \ldots, y_s) + w\rho(\delta_1, \delta_2, \ldots, \delta_s) \\
\text{s.t.} \quad & Ax = b \\
& B_s x + C_s y_s + \delta_s = e_s, \quad s \in \theta \\
& x \ge 0, \quad y_s \ge 0, \quad \delta_s \ge 0, \quad s \in \theta,
\end{aligned} \tag{28}$$

where δ_s is the deviation associated with the corresponding scenario s. In the objective function of this model, the first part stands for the measure of the solution robustness, and the second part means deviation from the constraint, that is, the measure of the model robustness.

There are several different forms of σ. In this paper, we use a form of σ which was proposed by Yu and Li [23]:

$$\sigma = \sum_{s\in\theta} p_s \cdot \xi_s + \lambda \sum_{s\in\theta} p_s \cdot \left| \xi_s - \sum_{s\in\theta} p_s \cdot \xi_s \right|, \tag{29}$$

where p_s is the probability of the scenario s and $\sum_{s\in\theta} p_s = 1$.

Under this form and the above robust model, we can get a robust formulation of model (26):

$$\begin{aligned}
\max \quad & \sum_{s\in\theta} p_s \cdot \pi_s - \lambda \sum_{s\in\theta} p_s \cdot \left| \pi_s - \sum_{s\in\theta} p_s \cdot \pi_s \right| \\
& - \sum_{s\in\theta} p_s \sum_{n=1}^{N}\sum_{i=0}^{T-1}\sum_{j=i+1}^{T} w_{ijn} \left| D_{ijn}^s - L_{ijn} \right| \\
\text{s.t.} \quad & \sum_{i=0}^{t-1}\sum_{j=t}^{T} L_{ijn} + \sum_{j=t+1}^{T} L_{tjn} - \sum_{i=0}^{t-1} L_{itn} \le L_n, \\
& t = 1, 2, \ldots, T-1, \quad n = 1, 2, \ldots, N; \\
& \sum_{j=1}^{T} L_{0jn} \le L_n, \quad n = 1, 2, \ldots, N; \\
& \sum_{n=1}^{N} L_n \le L; \\
& L_{ijn} \le \max\left\{ D_{ijn}^s \right\}, \\
& n = 1, 2, \ldots, n, \quad 0 \le i < j \le T, \quad s \in \theta \\
& L_{ijn} \in N \cup \{0\}, \quad n = 1, 2, \ldots, N, \ 0 \le i < j \le T; \\
& \pi_s = \sum_{n=1}^{N}\sum_{i=0}^{T-1}\sum_{n=1}^{N} r_n^s \cdot (j-i) L_{ijn}, \quad s \in \theta \\
& \lambda \ge 0, \quad w_{ijn} \ge 0,
\end{aligned} \tag{30}$$

where λ and w_{ijn} are nonnegative weighting parameters. In the objective function of this model (30), the first part is the expected revenue, while the second term is the mean absolute deviation of the revenue. We can regard parameter λ as a risk trade-off factor between expected revenue and deviation. The absolute deviation in the third term is a model robustness measurement while the parameters w_{ijn} are the penalty weights for the constraints violations.

In model (30), there is uncertain variable D_{ijn} in both objective function and constraints. In this paper, we use factor-based demand model. See and Sim [24] proposed a form of uncertain variable which can be written as this: $D_{ijn} \triangleq D_{ijn}^0 + \sum_{k=1}^K D_{ijn}^k z_k$, where $z \triangleq \{z_1, \ldots, z_K\}$, and the demand for each level is affinely dependent on uncertain factor z_k : $k = 1, \ldots, K$; K represents the number of such factors used to model demand. With the assumption U and theorems in Ang et al. [25], we can get the equivalent form of the constraint $D_{ijn}^s - L_{ijn} + \delta_{ijn}^s \geq 0$ as follows:

$$\begin{aligned}
\max \quad & \sum_{k=1}^{K} D_{ijn}^{s,k} z_k \\
\text{s.t.} \quad & \sum_{k=1}^{K} \left(\overline{D}_{ijn}^{s,k} \alpha_{k,1} + D_{ijn}^{s,k} \beta_{k,1} \right) \geq L_{ijn} - D_{ijn}^{s,0} - \delta_{ijn}^s, \quad (31) \\
& \qquad n = 1, 2, \ldots, N, \ 0 \leq i < j \leq T \\
& \alpha_{k,1} \geq 0, \quad \beta_{k,1} \geq 0, \quad k = 1, \ldots, K.
\end{aligned}$$

We can use a similar method to deal with the other constraint containing D_{ijn}. Finally, model (30) can be transformed in a linear term as follows:

$$\begin{aligned}
\max \quad & \sum_{s\in\theta} p_s \pi_s - \lambda \sum_{s\in\theta} p_s \left(\pi_s - \sum_{s\in\theta} p_s \pi_s + 2\varepsilon^s \right) \\
& - \sum_{s\in\theta} p_s \sum_{i=0}^{T-1} \sum_{j=i+1}^{T} \sum_{n=1}^{N} w_{ijn} \\
& \qquad \cdot \left(D_{ijn}^{s,0} + \sum_{k=1}^{K} D_{ijn}^{s,k} z_k \right. \\
& \qquad \left. - L_{ijn} + 2\delta_{ijn}^s \right) \\
\text{s.t.} \quad & \pi_s - \sum_{s\in\theta} p_s \cdot \pi_s + \varepsilon^s \geq 0 \\
& \sum_{k=1}^{K} \left(\overline{D}_{ijn}^{s,k} \alpha_{k,1} + D_{ijn}^{s,k} \beta_{k,1} \right) \geq L_{ijn} - D_{ijn}^{s,0} - \delta_{ijn}^s, \\
& \qquad n = 1, 2, \ldots, N, \quad 0 \leq i < j \leq T \\
& \sum_{i=0}^{t-1} \sum_{j=t}^{T} L_{ijk} + \sum_{j=t+1}^{T} L_{tjn} - \sum_{i=0}^{t-1} L_{itn} \leq C_k, \\
& \quad t = 1, 2, \ldots, T-1, \quad n = 1, 2, \ldots, N; \\
& \sum_{j=1}^{T} L_{0jn} \leq L_n, \quad n = 1, 2, \ldots, N; \\
& \sum_{n=1}^{N} L_n \leq L; \\
& \sum_{k=1}^{K} \left(\overline{D}_{ijn}^{s,k} \alpha_{k,2} + D_{ijn}^{s,k} \beta_{k,2} \right) \geq L_{ijn} - D_{ijn}^{s,0}, \\
& \qquad n = 1, 2, \ldots, N, \ 0 \leq i < j \leq T \\
& \pi_s = \sum_{n=1}^{N} \sum_{i=0}^{T-1} \sum_{n=1}^{N} r_n^s \cdot (j - i) L_{ijn}, \quad s \in \theta \\
& L_{ijn} \in N \cup \{0\}, \quad n = 1, 2, \ldots, N, \ 0 \leq i < j \leq T \\
& \lambda, w_{ijn}, \varepsilon^s, \delta_{ijn}^s \geq 0, \\
& n = 1, 2, \ldots, N, \quad 0 \leq i < j \leq T, \quad s \in \theta \\
& \alpha_{k,1}, \alpha_{k,2}, \beta_{k,1}, \beta_{k,2} \geq 0, \quad k = 1, \ldots, K.
\end{aligned} \quad (32)$$

The prominent feature of formulation (32) is that it is now in a linear programming form and ready to be solved by popular linear modeling packages like LINGO when the weighting parameters are assigned by the decision makers.

4.2. Illustrative Examples and Analysis. Consider such a 3PW company which can provide three different levels of storage service for the storage customers. According to the history data, there are three main demand scenarios s_1, s_2, and s_3; each demand scenario stands for a market condition. For example, s_1 may mean the market demand is high, and the storage demand of the three levels is large. s_2 means the storage market is bad, and the demand of each storage level is small. s_3 is the common condition, and the storage demand is medium. Suppose the probability of these three scenarios is 0.2, 0.2, and 0.6. The planning horizon is set to be 3 periods, and the total capacity of this warehouse is 2000 units. Demands for all pairs are forecast as shown in Tables 4–6.

In these tables, the "ST" means the storage time and "RT" means the retrieval time of the goods. The first number in Table 4 means that the demand which is stored in period 0 and retrieve in period 1 is 120. Suppose that, under multiple demand scenarios, the unit price of level 1 for each scenario is 20, 15, and 18 dollars per period, the unit price of level 2 is 30, 26, and 28 dollars per period, and the unit price of level 3 is 38, 32, and 36 dollars per period. For simplicity, all weights w_{ijn} are set to be equal to 1. By our robust model (32), we can get the optimal capacity allocation policies summarized in Table 7.

The total revenue of the 3PW is 219357 dollars. According to solution the linear programming model, the capacity allocation for level 1 is 951 units, the capacity allocation for level 2 is 612 units, and the capacity allocation for level 3 is 437 units. The optimal capacity allocation policies are summarized in Table 7.

Table 4: Demands of customers (scenario s_1).

ST: i	RT: j (level 1)				RT: j (level 2)				RT: j (level 3)			
	1	2	3	4	1	2	3	4	1	2	3	4
0	200	180	270	150	120	130	190	140	100	180	70	30
1	0	350	280	120	0	200	150	135	0	95	110	50
2	0	0	280	100	0	0	160	150	0	0	130	230
3	0	0	0	200	0	0	0	120	0	0	0	50

Table 5: Demands of customers (scenario s_2).

ST: i	RT: j (level 1)				RT: j (level 2)				RT: j (level 3)			
	1	2	3	4	1	2	3	4	1	2	3	4
0	100	140	290	100	90	100	120	110	120	100	70	40
1	0	320	180	120	0	150	200	120	0	80	100	40
2	0	0	220	90	0	0	120	100	0	0	100	150
3	0	0	0	150	0	0	0	100	0	0	0	70

Table 6: Demands of customers (scenario s_3).

ST: i	RT: j (level 1)				RT: j (level 2)				RT: j (level 3)			
	1	2	3	4	1	2	3	4	1	2	3	4
0	130	200	250	120	100	120	170	120	90	150	90	50
1	0	270	310	100	0	180	180	150	0	100	75	50
2	0	0	240	120	0	0	150	120	0	0	110	200
3	0	0	0	180	0	0	0	120	0	0	0	60

Table 7: Capacity allocation policy with multiple demands.

ST: i	RT: j (level 1)				RT: j (level 2)				RT: j (level 3)			
	1	2	3	4	1	2	3	4	1	2	3	4
0	170	180	270	110	120	120	185	125	120	80	90	50
1	0	270	250	100	0	195	180	140	0	120	110	50
2	0	0	200	110	0	0	150	135	0	0	130	230
3	0	0	0	200	0	0	0	120	0	0	0	70

According to Table 7, we can obverse that the demand of level 3 should be met firstly. That is because the unit revenue of level 3 is the highest; therefore the capacity allocation for level 3 can meet all the demand of level 3 for all scenarios. On the contrary, capacity allocation for level 1 is smaller than its market demand as a result of its lower unit revenue. 3TW company managers can improve their revenue by applying this optimization method.

5. Conclusions and Future Directions

In this paper, we consider the capacity allocation problem in 3PW company which provides several different level storage services in different price under uncertain market demand. On the revenue management perspective, we propose the mathematical formulations of this problem for both single and multiple periods condition. For the single period situation, as the demand is uncertain, we use three robust methods, absolute robust, deviation robust, and relative robust, to maximize the whole revenue. Based on the analysis of the optimal solution in each situation, we adapt continuous knapsack method to give the corresponding algorithm. Then we use some numerical examples to verify the practical applicability of our method. And we find that the 3PW company managers should provide the maximum possible units of the storage service level with high priority. As the objective function of each method is different, these three methods do not perform the same under the same market scenario. We find that the absolute robust method performs better than the other two methods in most situations. For the multiperiod situation, we propose a stochastic model for the multiperiod revenue management problem of the warehouse. A novel robust optimization technique is applied in this model to maximize the whole revenue. Then we give some numerical examples to verify the practical applicability of our method. The major contribution of this paper is that we use robust optimization to deal with the uncertainty of market demand in 3PW industry. In many of existing references of 3PW revenue management, authors consider revenue optimization under deterministic demand or suppose stochastic demand with known distribution such as Poisson process. In this paper, we do not know the distribution of market demand

in 3PW industry and linearize the uncertain mathematical programming by different robust methods.

There remain several limitations in our work. First, we consider the demand of each level and each period is independent. Actually, the demands between different levels may affect each other in some conditions and the demand in one period may be affected by its demand in last period. In our future research, we can analyze the affecting factors of demand and characterize the form of demand to improve the match degree between capacity allocation and demand. Secondly, we set the price of each level as exogenous variables in this paper. In the following research, we can combine the dynamic pricing policy and capacity allocation to improve the revenue of 3PW more efficiently.

Conflict of Interests

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

Acknowledgments

The authors would like to thank the Associate Editor and anonymous referees for their constructive comments and suggestions, which have led to significant improvements of the paper. This research is partially supported by National Natural Science Foundation of China (nos. 71131004, 71471071) and Humanities and Social Sciences Foundation of Chinese Ministry of Education (no. 12YJC630149).

References

[1] J. I. McGill and G. J. van Ryzin, "Revenue management: research overview and prospects," *Transportation Science*, vol. 33, no. 2, pp. 233–256, 1999.

[2] C. P. Wright, H. Groenevelt, and R. A. Shumsky, "Dynamic revenue management in airline alliances," *Transportation Science*, vol. 44, no. 1, pp. 15–37, 2010.

[3] K. T. Talluri and G. J. Van Ryzin, *The Theory Anpractice of Revenue Management*, Kluwer Academic Publishers, Boston, Mass, USA, 2004.

[4] M. K. Geraghty and E. Johnson, "Revenue management saves National Car Rental," *Interfaces*, vol. 27, no. 1, pp. 107–127, 1997.

[5] R. G. Cross, *Revenue Management*, Broadway Books, Bantam, Doubleday, Dell Publishing Group, New York, NY, USA, 1998.

[6] G. R. Bitran and S. M. Gilbert, "Managing hotel reservations with uncertain arrivals," *Operations Research*, vol. 44, no. 1, pp. 35–49, 1996.

[7] W. C. Chiang, J. C. Chen, and X. Xu, "An overview of research on revenue management: current issues and future research," *International Journal of Revenue Management*, vol. 1, no. 1, pp. 97–128, 2007.

[8] Y. Gong and R. B. M. de Koster, "A review on stochastic models and analysis of warehouse operations," *Logistics Research*, vol. 3, no. 4, pp. 191–205, 2011.

[9] F. Y. Chen, S. H. Hum, and J. Sun, "Analysis of third-party warehousing contracts with commitments," *European Journal of Operational Research*, vol. 131, no. 3, pp. 603–610, 2001.

[10] C. Lin, "Capacity allocation policy of third party warehousing with dynamic optimization," in *Proceedings of the 5th International Symposium on Advances in Computation and Intelligence (ISICA '10)*, vol. 6382 of *Lecture Notes in Computer Science*, pp. 297–303, Springer, Wuhan, China, 2010.

[11] X. Zhang, Y. Gong, and S. van de Velde, "Storage scheduling decision models for revenue management of self-storage warehouses," in *Proceedings of the 10th International TRAIL in Perspective Congress*, 2008.

[12] Y. Gong, R. B. M. de Koster, J. B. G. Frenk, and A. F. Gabor, "Increasing the revenue of self-storage warehouses by facility design," *Production and Operations Management*, vol. 22, no. 3, pp. 555–570, 2013.

[13] A. L. Soyster, "Technical note—convex programming with set-inclusive constraints and applications to inexact linear programming," *Operations Research*, vol. 21, no. 5, pp. 1154–1157, 1973.

[14] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, no. 4, pp. 769–805, 1998.

[15] A. Ben-Tal and A. Nemirovski, "Robust solutions of linear programming problems contaminated with uncertain data," *Mathematical Programming*, vol. 88, no. 3, pp. 411–424, 2000.

[16] A. Ben-Tal and A. Nemirovski, "On polyhedral approximations of the second-order cone," *Mathematics of Operations Research*, vol. 26, no. 2, pp. 193–205, 2001.

[17] L. El Ghaoui and H. Lebret, "Robust solutions to least-squares problems with uncertain data," *SIAM Journal on Matrix Analysis and Applications*, vol. 18, no. 4, pp. 1035–1064, 1997.

[18] L. El Ghaoui, F. Oustry, and H. Lebret, "Robust solutions to uncertain semidefinite programs," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 33–52, 1999.

[19] D. Bertsimas, D. Pachamanova, and M. Sim, "Robust linear optimization under general norms," *Operations Research Letters*, vol. 32, no. 6, pp. 510–516, 2004.

[20] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004.

[21] G. L. Vairaktarakis, "Robust multi-item newsboy models with a budget constraint," *International Journal of Production Economics*, vol. 66, no. 3, pp. 213–226, 2000.

[22] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, "Robust optimization of large-scale systems," *Operations Research*, vol. 43, no. 2, pp. 264–281, 1995.

[23] C. S. Yu and H. L. Li, "A robust optimization model for stochastic logistic problems," *International Journal of Production Economics*, vol. 64, no. 1, pp. 385–397, 2000.

[24] C.-T. See and M. Sim, "Robust approximation to multiperiod inventory management," *Operations Research*, vol. 58, no. 3, pp. 583–594, 2010.

[25] M. Ang, Y. F. Lim, and M. Sim, "Robust storage assignment in unit-load warehouses," *Management Science*, vol. 58, no. 11, pp. 2114–2130, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 186970, 15 pages
http://dx.doi.org/10.1155/2015/186970

Research Article

To Make Good Decision: A Group DSS for Multiple Criteria Alternative Rank and Selection

Chen-Shu Wang,[1] Heng-Li Yang,[2] and Shiang-Lin Lin[2]

[1]*Graduate Institute of Information Management, National Taipei University of Technology, Taiwan*
[2]*Department of Management Information Systems, National Chengchi University, Taiwan*

Correspondence should be addressed to Shiang-Lin Lin; shiang0623@gmail.com

Received 26 September 2014; Accepted 7 January 2015

Academic Editor: Jianming Shi

Decision making is a recursive process and usually involves multiple decision criteria. However, such multiple criteria decision making may have a problem in which partial decision criteria may conflict with each other. An information technology, such as *the decision support system* (DSS) and group DSS (GDSS), emerges to assist decision maker for decision-making process. Both the DSS and GDSS should integrate with a symmetrical approach to assist decision maker to take all decision criteria into consideration simultaneously. This study proposes a GDSS architecture named hybrid decision-making support model (HDMSM) and integrated four decision approaches (Delphi, DEMATEL, ANP, and MDS) to help decision maker to rank and select appropriate alternatives. The HDMSM consists of five steps, namely, criteria identification, criteria correlation calculation, criteria evaluation, critical criteria selection, and alternative rank and comparison. Finally, to validate the proposed feasibility of the proposed model, this study also conducts a case study to find out the important indexes of corporate social responsibility (CSR) from multiple perspectives. As the case study demonstrates the proposed HDMSM enables a group of decision makers to implement the MCDM effectively and help them to analyze the relation and degree of mutual influence among different evaluation factors.

1. Introduction

People are making decision all the time. Typically, decision making is a recursive process in which decision maker may repeatedly move back and forth among multiple decision steps, such as objective clarification, decision criteria identification, alternative rank, and selection. For decision maker, the primary concern is to pick up an appropriate choice from a group of candidate alternatives [1]. Such decision making process usually involves multiple decision criteria [2], so called as multiple criteria decision making (MCDM) [3]. Lots of researches are devoted to resolve sorts of MCDM problem. For example, Hung et al. provide a novel MCDM approach to solve the knowledge management (KM) adoption problem and rank the gaps of the KM aspects in control items to achieve the aspired level of performance. The findings demonstrate that the KM gaps within the service industry are higher than the gaps within the integrated circuit and banking industries [4]. Also, Hsu et al. combine DEMATEL on ANP with VIKOR to solve the recycled materials vendor selection problems of multiple dimensions and criteria that are interdependent, instead of the independent assumption of an analytic hierarchy process, for mimicking the real world [5]. Besides, Chiu et al. focus on assessing e-store strategies to reduce the gaps in the resulting customer satisfaction and combine several multicriteria decision methodologies to conduct three real cases [6]. As mentioned above, MCDM is a complicated problem but can reflect real world precisely and therefore we should pay more attention to resolve MCDM issue.

However, unfortunately, most people are much poorer at decision making than they think. For illustration, there is a misconception that the decision maker thought they do not have enough information to make good decision [7]. Contrarily, in most cases, they spend much time to collect relative (or even irrelevant) information and trap themselves

in the huge amount of information. Decision making is a sophisticated art and decision makers indeed require some help.

An information technology, as known as decision support system (DSS), emerges to assist decision maker to accelerate the convergence of decision-making process. DSS is interactive computer-based information system which helps decision-makers utilize knowledge base and models to solve ill-structured problems [8]. For these decades, DSS has been widely applied among domains as follows. Koo et al. developed a DSS based on case-based reasoning approach for determining the optimal size of new expressway service areas [9]. Gottschlich and Hinz proposed a DSS design that enables investors to include the crowd's recommendations in their investment decisions and use it to manage a portfolio [10]. Hu and Sheng also proposed a DSS for public logistics information service management and optimization for vehicle drivers and owners, logistics customers, and related logistics service providers and management institutes [11]. Then, to respond accordingly to the requirement of group decision making, DSS further evolves into group DSS (GDSS) to help group of decision makers with efficiency decision making [12–16]. A group decision support system (GDSS) is a hybrid system that uses an elaborate communications infrastructure and quantitative models to help a team of decision makers solve problems and make choices [17, 18]. However, the decision process is a classical multiple criteria problem that partial decision criteria may conflict with each other. Both the DSS and GDSS should integrate with a symmetrical approach to assist decision maker to take all decision criteria into consideration simultaneously.

In this research, we proposed a hybrid GDSS architecture, named HDMSM, integrated four decision approaches (Delphi, DEMATEL, ANP, and MDS) to help decision maker with alternative rank and selection issue. HDMSM consists of four steps. In Step 1, HDMSM adopts Delphi to collect the decision criteria from domain experts. After that, in Step 2, domain experts use DEMATEL approach to evaluate the relevant among the selected criteria (in Step 1) and then generate a correlation matrix of these decision criteria. Then, in Step 3, HDMSM adopts ANP to calculate the correlation and important weight for each decision criteria in Step 4. Finally, in Step 5, the MDS can rank all available alternatives according to these important weights and visualize the similarity (or difference) of all available alternatives.

The priorities of decision criteria imply the preference of domain expert and therefore, to make better decision, and decision maker can make their choice according to the alternative rank via HDMSM. Also, the visual abilities of HDMSM enable decision maker to compare all available alternative form perspective and then improve decision making quality. Finally, we provide a system demonstration section to illustrate that how HDMSM aggregated the opinion from a group of domain experts. How to appropriately integrate a variety of MCDM approaches is an important issue in decision science [19]. HDMSM provides a valuable recommendation for decision maker to optimize multiple criteria decision making.

The remainder of this paper is organized as follows. Section 2 briefly reviews four decision analysis methodologies adopted in this study. Section 3 presents the proposed hybrid decision-making support model (HDMSM) and details the operational process of HDMSM. To validate the feasibility of HDMSM, according to five steps of HDMSM, a case study in Section 4 illustrates how decision makers appropriately select multiple decision criteria for corporate social responsibility (CSR) implementation step by step. Finally, Section 5 concludes some interest finding and proposes possible future research opportunities.

2. Literature Review

The HDMSM proposed in this study integrates four decision approaches, namely, Delphi, DEMATEL, ANP, and MDS, to help decision maker with criteria selection and alternative ranking when facing a decision problem. Four methodologies are briefly introduced below.

2.1. Delphi Method. Delphi method relies mainly on a panel of experts' experiences, intuition, and value judgment. The experts participate in multiple rounds of questionnaire interviews and are given ways to understand one another's viewpoints on the same question. The experts are encouraged to revise their previous opinions, so that the experts as a group can finally reach a consensus on the goal of decision making [19]. To perform the Delphi method, the following procedures are included.

2.1.1. Choose a Panel of Decision-Making Experts and Select the Criteria for Decision Making. Determine the goal of decision making and list relevant evaluation criteria for the decision making, choose experts in the related field to form an expert group, and invite the experts to answer in a Delphi expert questionnaire interview for multiple rounds. The experts must judge the importance and suitableness for each evaluation criteria and give each criterion a score between 0 and 100.

2.1.2. Test the Expert Group Consensus. To enable expert group to gradually reach a general agreement of opinion, a consensus deviation index (CDI) for each evaluation criteria is calculated as a round of the Delphi expert questionnaire finished. A smaller CDI indicates a higher consensus among the experts. In general, a CDI threshold is set to 0.05. That is, when the last round of Delphi expert questionnaire is completed and the CDI for all of the evaluation criteria is smaller than 0.05, this indicates a consensus of experts has been reached [20]. Herein, the score of the jth criterion rated by the hth expert in the tth round of Delphi questionnaire is defined as X_{jht}, and the CDI is calculated by the formula:

$$\mathrm{CDI}_{jt} = \frac{S_{jt}}{\max_j \overline{X}_{jt}}, \quad \forall j, t, \tag{1}$$

where $\overline{\overline{X}}_{jt}$ is the mean of the scores of the jth criterion rated by all the experts in the tth round of Delphi expert

questionnaire, and S_{jt} is the standard deviation of the scores of the jth evaluation criterion rated by all the experts in the tth round of Delphi expert questionnaire.

2.1.3. Normalize the CDI and Choose the Evaluation Criteria. The CDI calculated based on the last round of Delphi expert questionnaire has to be normalized to derive the relative weight W_j of each of the evaluation criteria. A small weight indicates the criterion does not have sufficient influence on the decision problem. In other words, when the weight of a particular criterion is less than a threshold set by the decision maker, the criterion is deleted from the candidate decision criteria, wherein W_j is calculated by the following formula:

$$W_j = \frac{\overline{X}_j}{\sum_{j=1}^{M} \overline{X}_j}, \quad \forall j. \tag{2}$$

The evaluation criteria involved in decision making problem can be identified via Delphi. Then, DEMATEL is adopted to analyze the direct/indirect effects among these evaluation criteria detailed below.

2.2. Decision-Making Trial and Evaluation Laboratory. Decision making trial and evaluation laboratory (DEMATEL) was originated from the Geneva of the Battelle Memorial Institute in 1973. It effectively observes the level of mutual influence among different factors and understands the complicated cause-and-effect relationship in the decision problem [21]. The analytic processes are listed below.

2.2.1. Define the Correlation among Evaluation Factors. List the factors that may affect the decision-making problem through literature review or brainstorming and interview with the domain experts to determine the correlation between any two factors.

2.2.2. Establish Direct Relation Matrix. As the decision problem has n evaluated factors, an $n * n$ direct relation matrix showing the scores of influencing degree is established, which is presented as *Z-matrix in formula (3)*. Element z_{ij} represents the degree by which the factor z_i affects factor z_j:

$$Z = \begin{array}{c} \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{bmatrix} 0 & z_{12} & \cdots & z_{1n} \\ z_{21} & 0 & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & 0 \end{bmatrix} \end{array}. \tag{3}$$

2.2.3. Establish Direct/Indirect Relation Matrix. In order to understand whether two evaluation criteria relate to each other indirectly, formula (4) produces a direct/indirect relation matrix T, where I is the identity matrix:

$$T = \left| X (I - X)^{-1} \right|. \tag{4}$$

2.2.4. Calculate the Prominence Score. If t_{ij} is an element of matrix T, where $i, j = 1, 2, \ldots, n$, the sum of the column and the row are denoted by D_i and R_j, respectively. Among them, D_i represents the sum of the criterion i influencing other criteria, R_j represents the sum of the criterion j being affected by other criteria, and $D + R$ represents the prominence degree of each criterion in the decision-making problem. A prominence score can reveal both the importance and mutual effects among these criteria. In other words, DEMATEL analyzes the direct and indirect effect of these evaluation criteria on decision problem. The analysis results can be further plotted as a network structure via ANP approach.

2.3. Analytic Network Process. ANP is a decision-making analytical method that uses network and nonlinear structure to represent a decision problem. ANP is developed in response to the fact that many decision problems in the realistic environment could not be presented with the structured hierarchy. The main objective of ANP is to correct the traditional AHP, with which the problems of dependence and feedback might occur between the criteria or the layers [22].

ANP can decompose a decision problem into multiple types of dimensions, and each dimension can include multiple criteria. The dimensions and the criteria are correlated with one another to form a network structure of the evaluation framework, and arrows are used to indicate their mutual influence.

Through the pairwise comparison of among each two criteria, ANP is calculated to acquire the eigenvectors of criteria and to form a Supermatrix, as shown in

$$W = \begin{bmatrix} W_{i_1 j_1} & W_{i_1 j_2} & \cdots & W_{i_1 j_n} \\ W_{i_2 j_1} & W_{i_2 j_2} & \cdots & W_{i_2 j_n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{i_m j_1} & W_{i_m j_2} & \cdots & W_{i_m j_n} \end{bmatrix}. \tag{5}$$

Through normalization of the Supermatrix and complex matrix multiplication, a limit supermatrix containing the weights of the evaluation criteria can be obtained. According to these weights, the decision maker can figure out the priority of evaluation criterion for decision problem solving.

2.4. Multidimensional Scaling. Multidimensional scaling (MDS) is a data reduction method, which uses the distance or similarity between data points to locate the spatial coordinates and the relative positions of several given data in the low-dimensional space [23].

MDS computes the Euclidean distance between each two factors and shows all factors in perceptual map which has two dimensions. If the similarity between two factors is more stronger, the configuration of two factors would be more close in the map. As a result, through the illustration of perceptual map, the spatial relation among factors can be visualized more clearly. The classification results of factors can be achieved via spatial difference analysis that helps decision maker to easily grasp the concept of factors.

To obtain the perceptual map, the Euclidean distance (D_{ij}) between each two factors should be computed first. Further, the Euclidean distance matrix of the factors is generated. The Euclidean distance equation is shown as

$$D_{ij} = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \left(x_{i2} - x_{j2}\right)^2 + \cdots + \left(x_{ik} - x_{jk}\right)^2}, \quad (6)$$

where x_i denotes the perceived value of factor a and x_j denotes the coordinate of factor b.

For decision making, decision maker must locate important decision criteria and evaluate the fitness of all possible alternatives. To increase decision process, decision maker needs to compare these alternatives as soon as possible. Via MDS, the visualization of candidate alternative enables decision maker to quickly grasp the similarities and dissimilarities among the alternatives.

3. Hybrid Decision-Making Support Model (HDMSM)

This study proposes a GDSS architecture named hybrid decision-making support model (HDMSM) as shown in Figure 1. According to the decision-making procedures, HDMSM consists of five steps, namely, criteria identification, criteria correlation calculation, criteria evaluation, critical criteria selection, and alternative rank and comparison, which are detailed below.

Step 1 (criteria identification). To make MCDM, some appropriate evaluation dimensions and criteria are selected as candidate decision criteria. Then, a domain expert group is organized via Delphi for conducting multiple rounds of questionnaire interviews. To finally establish a consensus score for each of the criteria, the experts group gradually reaches a consensus in their opinions. Based on the consensus scores, the top n decision criteria that with the highest important weight are chosen from the candidate criteria as evaluation criteria. Finally, these decision criteria are, respectively, given a detailed definition according to the opinions from the interviewed expert group.

Step 2 (criteria correlation calculation). To understand the correlation among the selected evaluation criteria in Step 1, it is necessary to further create a correlation matrix. According to the prominence of these criteria, for each criterion, DEMATEL evaluates the correlation among these criteria and calculates the prominence score.

Step 3 (criteria evaluation). To show the direct relation among these evaluation criteria, HDMSM plots a network structure of the evaluation framework. And then, an ANP expert questionnaire is designed based on the plotted evaluation framework. The domain experts fulfilled the ANP questionnaire and the collected questionnaires are further analyzed via ANP method to derive the absolute weight. Thereafter, for each decision criterion, it cross-multiplies the absolute weight by the consensus score (in Step 1) and the prominence score (in Step 2), to obtain a composite important weight. Finally, according to the composite importance weights, HDMSM can rank the decision criteria for MCDM.

Step 4 (critical criteria selection). To evaluate the feasibility for each alternative, multiple decision criteria are usually taken into consideration simultaneously. Therefore, in this step, according to the analysis results in Step 3, a decision maker sets a threshold for the composite important weight. If the evaluation criteria with a higher composite important weight, decision maker can pick them out and conduct pairwise comparison with alternatives to find out the priority of each alternative.

Step 5 (alternative rank and comparison). For particular complex case, appropriate alternative might be a combination of several alternatives. Therefore, the decision maker must understand the effect of each of the evaluation criteria. Additionally, they need to analyze the structural similarity among different alternatives. Therefore, in Step 5, MDS analysis is used to analyze the results from Step 4 for allocating the positions of the alternatives in a perceptual map of second dimension, so as to present the similarity and dissimilarity among the alternatives through visualization.

HDMSM adopts four methodologies that complement each other. Delphi summarizes the opinion from expert group and then generates appropriate evaluation factors for a multicriteria decision making. DEMATEL reveals the correlation among these decision factors. ANP implements pairwise comparison of these factors and derives the important weights for all evaluation factors. By importance ranking, ANP provides decision maker with the insight into the decision problem. Based on the analysis results, MDS generates the perceptual map to improve the representation of alternative analysis. The visualization representation enables decision maker to quick-grasp the similarity and dissimilarity among alternatives and increase decision making process.

4. Case Study: To Select the Evaluation Factors of Corporate Social Responsibility

A case study is implemented based on the proposed HDMSM. In this case study, we intend to find out the important indexes of corporate social responsibility (CSR) from multiple perspectives. CSR involves the conduct of a business so that it is economically profitable, law abiding, ethical, and socially supportive. To be socially responsible, profitability and obedience to the law are foremost conditions to discuss the firm's ethics and the extent to which it supports the society in which it exists with contributions of money, time, and talent. Thus, CSR is composed of four dimensions: economic, legal, ethical, and philanthropic [24, 25] and the selection of the evaluation factor for each dimension would be a complicated issue. According to these four dimensions, some appropriate evaluation indexes are identified via Delphi method in Step 1, and the correlation among the evaluation indexes is calculated via DEMATEL method in Step 2. Then, in Step 3, HDMSM adopts ANP to evaluate the important

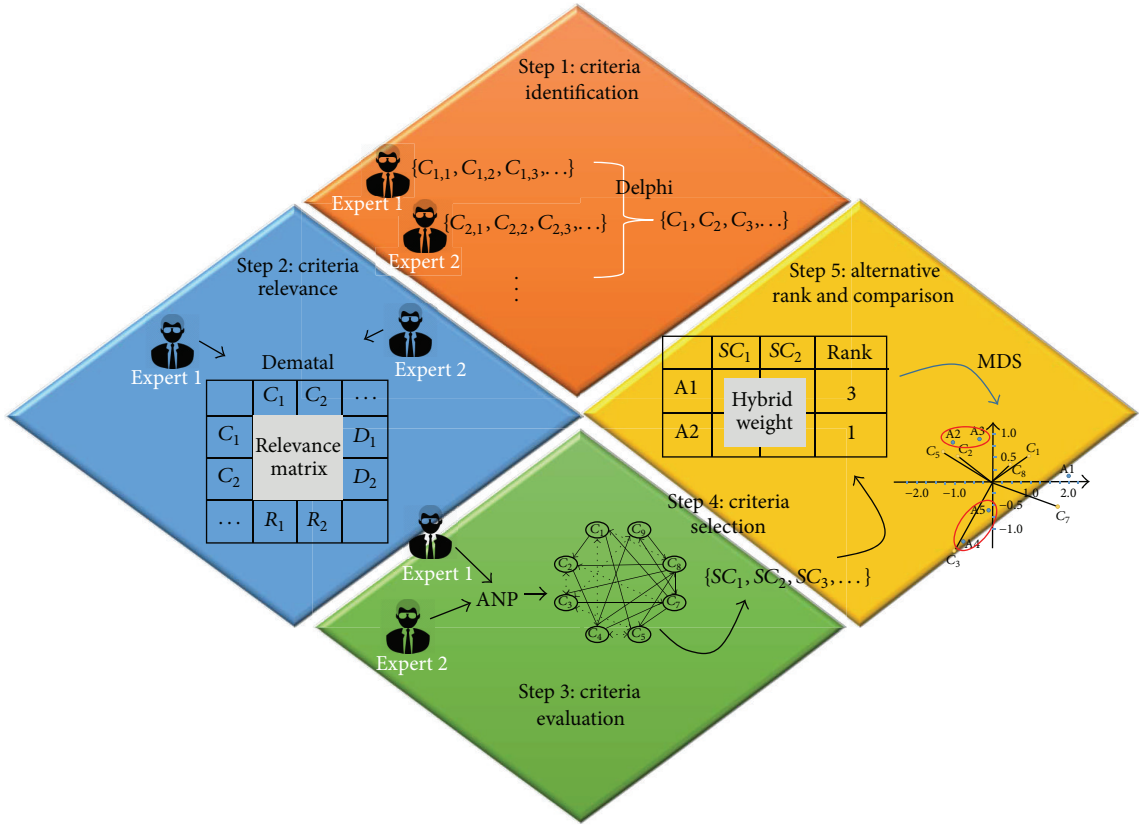


Figure 1: Hybrid decision-making support model (HDMSM).

ranking of these evaluation indexes for an enterprise to implement its corporate social responsibility and for different interested parties in connection with the enterprise. Finally, in Step 4, HDMSM selects 10 critical indexes through composite important weight and compares the similarity and dissimilarity among the interested parties via MDS in Step 5.

4.1. Criteria Identification. To make complicate multicriteria decision, like evaluation factors of CSR, it is necessary to select dimensions, criteria, and available alternatives as evaluation factors. Table 1 summarized some previous researches related to CSR. In total, there are 4 groups of evaluation dimensions and 28 evaluation indexes (i.e., evaluation criteria). Furthermore, five clusters of interested parties are chosen, namely, suppliers (A1), shareholders (A2), employees (A3), customers (A4), and the general public (A5), as the alternatives in this study.

To implement the case study, five domain experts were invited to form a decision-making team and fulfill a Delphi expert questionnaire that consists of 28 CSR evaluation criteria. The expert gave each criterion a score between 0 and 100 points. Then, for two discussion rounds, the experts gradually reached a consensus on the target decision problem (i.e., CDI < 0.05). According to the average score of decision criteria (the threshold is set 60-points), some criteria are deleted from the candidate pool of decision criteria. For the remained criteria, to uniform evaluation scale, the normalization process was conducted to obtain the rating weight for each CSR evaluation criteria as shown in Table 2. For illustration, some evaluation criteria (such as P1, P2, and L5) are more influential for business to solve CSR decision problem.

4.2. Establishment of Criteria Correlation. After the CSR evaluation criteria were identified via Delphi method in Step 1, DEMATEL expert questionnaire interview was implemented to determine the relation and the degree of mutual influence among the criteria to generate a direct relation matrix. The questionnaire scaling is 0 to 3 points in which 0 indicates no relation and 3 indicates a significant relation among the decision criteria [21]. A DEMATEL questionnaire sample is shown in Table 3.

4.3. Calculate Prominence of Criteria. After obtaining the direct relation matrix, formula (4) can be used to calculate and generate a direct/indirect relation matrix as shown in Table 4.

According to the direct/indirect relation matrix in Table 4, the sum of the column and the row is denoted by D_i and R_j, respectively. D_i represents the sum of the criterion i influencing other criteria, R_j represents the sum of the criterion j being affected by other criteria, and the

Table 1: CSR dimensions and criteria selection.

Dimensions		Evaluation criteria
Economic responsibilities	E1	Reasonable product prices
	E2	Transparent business operations
	E3	Avoid price collusion with any competitor
	E4	Stimulate local economic development
	E5	Increase employment opportunities
	E6	Maximum possible profits for the organization
	E7	Honour agreed-upon contracts
Legal responsibilities	L1	Implement the consumer protection act
	L2	Ensure good and safe working environment for employees
	L3	Provide employees with the newest information on pertinent laws
	L4	Provide occupational injury compensation and health insurance systems
	L5	Keep customer information confidential and protect it against illegal use
	L6	Provide proper waste disposal and reduce pollutant emissions
	L7	Actively inspect and certify products
Ethical responsibilities	M1	Provide consumers with customer complaint service and thorough follow-up service
	M2	Provide employees with ways of improving their working conditions
	M3	Eschew exaggerated or false advertisements
	M4	Pay salaries and wages on time
	M5	Provide employees with a good working environment
	M6	Provide employees with fair selection, promotion, termination, and retirement systems
	M7	Provide employees with reasonable welfare and protection
	M8	Provide employees with educational training and living-related assistance
	M9	Cooperate with the government in energy-saving and carbon-reduction policy
	M10	Ensure transparent production processes
Philanthropic responsibilities	P1	Protect vulnerable social groups
	P2	Engage in charitable activities
	P3	Provide for community welfare
	P4	Use company resources efficiently

prominence score $(D_i + R_i)$ of each criterion can be obtained. The prominence scores are shown in Table 5. As an evaluation criterion has a higher prominence score $(D_i + R_i)$, it indicates that the evaluation criterion has a higher prominence for decision-making problem. For example, in Table 5, the decision criteria M7 (providing employees with reasonable welfare and protection) with the highest prominence score is identified as the strongest influence decision criteria in the CSR case study.

4.4. Construct the Network Structure of Evaluation Framework. In the previous step, DEMATEL generated the direct relation among the CSR evaluation criteria, as shown in Table 3, and the direct relation was then used to plot a network structure of evaluation framework, as shown in Figure 2.

Next, an ANP expert questionnaire was designed based on the network structure of evaluation framework. Total 15 interviewers finished the ANP questionnaires. The 15 interviewers included 3 middle-high level managers from different companies. Taking economic responsibility dimension as an example, the pairwise comparisons for the criteria E1–E5 were conducted, and the 9-point paired-comparison scaling was used for rating [26] as shown in Table 6. The answers to the questionnaires must conform to the logic of transitivity. That is, for illustration, in a questionnaire, when E1 > E2 and E2 > E3, then the condition of E1 > E3 must exist. If not, the questionnaire is considered an invalid questionnaire.

4.5. Criteria Weighting and Priority Ranking. The collected questionnaires were further analyzed and processed using ANP to create an unweighted super matrix as shown in Table 7. The values in the matrix are the relative weights of the CSR evaluation criteria under each question in the ANP expert questionnaire. For instance, based on E3, the relative weights of L1 and M1 obtained from the pairwise comparison are 0.75 and 0.25, respectively.

We cross-multiplied the values in each row of the unweighted super matrix by the weight of the dimension, respectively, and then a weighted super matrix can be obtained as shown in Table 8. The weight of each dimension is determined according to the expert's feedback. In this case, the weights of the four dimensions of economic responsibility, legal responsibility, moral responsibility, and philanthropic responsibility are 0.25, 0.4, 0.2, and 0.15, respectively.

To get a uniform weighted scale of super matrix, the normalization is implemented. A limit super matrix is shown in Table 9. The normalized matrix (in Table 8) is multiplied by itself repeatedly to convert the values (by row) into a limit value and generate the limit super matrix, wherein the values of each row are the importance weights of the CSR evaluation criteria.

According to the above MCDM method, the importance weights or scores for three groups of the CSR evaluation criteria are derived. However, the use of the results from one single decision-making analytic method to determine the importance level of each of the evaluation criteria to the overall decision problem seems to be too arbitrary and less persuasive and might even cause bias in determining the target decision, which obviously deviates from the initial purpose of employing different decision-making analytic

Table 2: CDI and rating weights of CSR criteria.

(D)	(C)	Score of final round					σ	CDI	Means	Rating weight by normalization
		1	2	3	4	5				
Economic	E1	90	90	80	85	90	4.47	0.050	87.0	**0.053**
	E2	90	95	85	85	90	4.18	0.047	89.0	**0.054**
	E3	75	70	75	70	80	4.18	0.047	74.0	**0.045**
	E4	75	70	70	75	75	2.74	0.031	73.0	**0.045**
	E5	85	80	80	80	75	3.54	0.040	80.0	**0.049**
	E6	55	60	60	55	50	4.18	0.047	56.0	(deleted)
	E7	55	60	60	55	55	2.74	0.031	57.0	(deleted)
		Sum of nondeleted mean:							403.0	—
Legal	L1	85	80	85	90	85	3.54	0.038	85.0	**0.052**
	L2	75	85	80	80	77	3.78	0.041	79.4	**0.048**
	L3	75	78	70	80	80	4.22	0.046	76.6	**0.047**
	L4	70	75	72	78	75	3.08	0.034	74.0	**0.045**
	L5	90	95	90	90	95	2.74	0.030	92.0	**0.056**
	L6	85	87	90	93	95	4.12	0.045	90.0	**0.055**
	L7	75	70	80	76	73	3.70	0.040	74.8	**0.046**
		Sum of nondeleted mean:							571.8	—
Ethical	M1	85	80	83	88	90	3.96	0.044	85.2	**0.052**
	M2	61	62	55	54	55	3.78	0.042	57.4	(deleted)
	M3	55	54	50	45	53	4.04	0.045	55.0	(deleted)
	M4	70	65	68	73	70	2.95	0.033	69.2	**0.042**
	M5	45	53	52	50	47	3.36	0.037	49.4	(deleted)
	M6	70	77	75	73	80	3.81	0.042	75.0	**0.046**
	M7	65	63	60	70	66	3.70	0.041	64.8	**0.040**
	M8	45	48	45	52	50	3.08	0.034	48.0	(deleted)
	M9	90	93	85	88	95	3.96	0.044	90.2	**0.055**
	M10	85	88	90	93	90	2.95	0.033	89.2	**0.054**
		Sum of nondeleted mean:							473.6	—
Philanthropic	P1	98	90	95	95	100	3.78	0.040	95.6	**0.058**
	P2	95	90	92	96	95	2.51	0.026	93.6	**0.057**
	P3	55	60	56	58	55	2.17	0.023	56.8	(deleted)
	P4	61	63	55	54	55	4.10	0.043	57.6	(deleted)
		Sum of nondeleted mean:							189.2	**1.000**

D denotes dimension, C denotes criterion, σ denotes standard deviation, and CDI denotes consensus deviation index.

methods. To avoid the above condition and to upgrade the accuracy of results from decision-making analysis, this study further multiplied the importance weights of the criteria derived with Delphi and ANP by the prominence scores of the criteria found by DEMATEL. However, since some criteria have a prominence score less than 1 (i.e., $D_i + R_i < 1$) that would result in an even smaller value after multiplication, it is necessary to add 1 to each of the prominence scores before multiplying the importance weights by the prominence scores. Finally, a composite importance weight was obtained for each of the CSR evaluation criteria, and a priority ranking of all criteria was obtained according to their composite importance weights. The calculation results are shown in Table 10.

As shown in Table 10, the ranking of the criteria based on the composite importance weights is not exactly the same priority ranking for all three MCDM methods, and some criteria have quite different importance levels in different ways of ranking. For instance, referring to the prominence score of DEMATEL, E2 is the third important index. However,

Table 3: Direct relation matrix.

Criteria	E1	E2	E3	E4	E5	L1	L2	L3	L4	L5	L6	L7	M1	M4	M6	M7	M9	M10	P1	P2
E1	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E2	1	0	3	0	0	0	0	0	0	1	2	3	0	0	2	0	0	3	0	0
E3	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
E4	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L1	3	0	2	0	0	0	0	0	0	3	0	3	2	0	0	0	0	1	0	0
L2	0	0	0	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	0	0
L3	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0
L4	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	3	0	0	1	0
L5	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M4	0	0	0	0	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
M6	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
M7	0	0	0	0	0	0	3	3	3	0	0	0	0	1	2	0	0	0	0	0
M9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M10	0	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0	3	0	0	0
P1	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
P2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0

Table 4: Direct/indirect relation matrix.

Criteria	E1	E2	E3	E4	E5	L1	L2	L3	L4	L5	L6	L7	M1	M4	M6	M7	M9	M10	P1	P2
E1	0.00	0.01	0.21	0.14	0.03	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E2	0.07	0.01	0.22	0.01	0.01	0.03	0.02	0.00	0.00	0.07	0.18	0.25	0.02	0.00	0.14	0.00	0.04	0.20	0.00	0.00
E3	0.02	0.07	0.03	0.00	0.00	0.07	0.00	0.00	0.00	0.02	0.01	0.03	0.08	0.00	0.01	0.00	0.00	0.02	0.00	0.00
E4	0.00	0.00	0.00	0.01	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
E5	0.00	0.00	0.00	0.07	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L1	0.21	0.01	0.19	0.03	0.01	0.05	0.00	0.00	0.00	0.21	0.02	0.23	0.15	0.00	0.00	0.00	0.01	0.07	0.00	0.00
L2	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.01	0.14	0.00	0.07	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.00
L3	0.00	0.00	0.00	0.00	0.02	0.00	0.02	0.06	0.02	0.00	0.00	0.00	0.00	0.22	0.01	0.07	0.00	0.00	0.00	0.00
L4	0.00	0.00	0.00	0.01	0.02	0.00	0.26	0.05	0.08	0.00	0.02	0.00	0.00	0.02	0.03	0.22	0.00	0.00	0.07	0.01
L5	0.04	0.00	0.04	0.01	0.00	0.21	0.00	0.00	0.00	0.04	0.00	0.05	0.03	0.00	0.00	0.00	0.00	0.01	0.00	0.00
L6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
L7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M4	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.21	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.01	0.00	0.00	0.00	0.00
M6	0.00	0.00	0.00	0.00	0.07	0.00	0.14	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M7	0.00	0.00	0.00	0.00	0.02	0.00	0.28	0.24	0.25	0.00	0.02	0.00	0.00	0.12	0.14	0.07	0.00	0.00	0.02	0.00
M9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
M10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.20	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00
P1	0.00	0.00	0.00	0.08	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.14
P2	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.21	0.03

if taken the composite weight into consideration, the same index of E2 is the eleventh. For another example, according to the Delphi weight, P1 and P2 are the top two important CSR evaluation criteria but only rank last two in the priority ranking based on the composite weight. As demonstration, both the analysis results of all MCDM methodologies and composite weights have the reference values.

4.6. Alternative Comparison. In general, there are more than one alternative for decision maker to choose for problem solving. For each alternative, they may have levels of effect on different evaluation criteria. Therefore, according to described analytic results, in the study, the importance weight threshold is set to 0.05 for top 10 CSR critical criteria selection. And then, a second pairwise comparison is

Table 5: Prominence of CSR evaluation criteria.

CSR evaluation criteria		R_i	D_i	$D_i + R_i$
E1	Reasonable product prices	0.35	0.43	0.78
E2	Transparent business operations	0.11	1.30	1.41
E3	Avoid price collusion with any competitor	0.58	0.36	0.94
E4	Stimulate local economic development	0.39	0.22	0.61
E5	Increase employment opportunities	0.82	0.08	0.90
L1	Implement the consumer protection act	0.38	1.03	1.41
L2	Ensure good and safe working environment for employees	0.85	0.31	1.16
L3	Provide employees with the newest information on pertinent laws	0.56	0.42	0.98
L4	Provide occupational injury compensation and health insurance systems	0.53	0.80	1.32
L5	Keep customer information confidential and protect it against illegal use	0.35	0.41	0.76
L6	Provide proper waste disposal and reduce pollutant emissions	0.53	0.00	0.53
L7	Actively inspect and certify products	0.67	0.00	0.67
M1	Provide consumers with customer complaint service and thorough follow-up service	0.29	0.00	0.29
M4	Pay salaries and wages on time	0.41	0.36	0.76
M6	Provide employees with fair selection, promotion, termination, and retirement systems	0.34	0.41	0.74
M7	Provide employees with reasonable welfare and protection	0.41	1.18	1.59
M9	Cooperate with the government in energy-saving and carbon-reduction policy	0.26	0.00	0.26
M10	Ensure transparent production processes	0.31	0.60	0.91
P1	Protect vulnerable social groups	0.34	0.47	0.81
P2	Engage in charitable activities	0.18	0.29	0.47

conducted for the five interested clusters (i.e., the alternatives used in this study). To facilitate the subsequent alternative similarity analysis, the 9-point paired-comparison scaling was also used for measurement. Finally, the relative weights of the interested clusters under the selected 10 critical criteria were obtained as shown in Table 11.

Based on Table 11, the Euclidean distance between any two of the interested clusters is further calculated and the Euclidean distance matrix is created as shown in Table 12. In the Euclidean distance matrix, the smaller the value between two alternatives (such as the interested parties), the closer the distance between them, meaning a higher similarity between two alternatives. Then, according to this matrix, the positions of the interested clusters in a two-dimensional coordinate system can be located, and a perceptual map of alternative allocation can be plotted.

From the perceptual map shown in Figure 3, the similarity and dissimilarity among the clusters (such as three elliptic areas in Figure 3) can be clarified clearly. For CSR example in Section 4, the perceptual map can visualize the evaluation criteria for CSR from different perspectives (such as A1 to A5) and provide decision maker with detail information. As shown in Figure 3, for illustration, we can find that the consideration of shareholder is similar to supplier and employee while business is intended to plan and implement CSR issues. Referring to the analysis result of the proposed HDMSM, decision maker can develop a solution that fulfilled the most correct decision-making policy.

5. Conclusions and Further Work

MCDM has been an important issue and many researches are devoted to help people make better decision. Particularly, some decision-making analytic methodologies (such as ANP and DEMTEAL) and information technologies (such as DSS and GDSS) can help decision makers to analyze the decision problem, collect information, indicate the available alternatives, and so on. However, the MCDM involves multiple decision criteria and, the worst, these criteria might mutually influence one another to lead to a complicated situation. Decision making is a sophisticated art and decision makers indeed require some help.

This study proposed a hybrid decision-making support model (HDMSM) that is an integrated GDSS architecture consisting of five steps, namely, criteria identification, criteria correlation, criteria evaluation, criteria selection, and alternative rank and comparison. Further, HDMSM appropriately integrates four systemic decision approaches (Delphi, DEMATEL, ANP, and MDS) to help the decision maker with alternative rank and selection issue. HDMSM consists of five steps. In Step 1, HDMSM adopts Delphi to collect and identify the decision criteria from domain experts. After that, in Step 2, domain experts use DEMATEL to evaluate the correlation among the selected criteria (in Step 1) and then generate a relation matrix of these decision criteria. Then, in Step 3, HDMSM adopts ANP to calculate the complicate correlation and importance weight for each decision criterion in Step 4. Finally, in Step 5, MDS can rank all available alternatives according to the importance weights and visualize the similarity (or difference) among all available alternatives.

A case study was implemented based on the proposed HDMSM. The case study intends to find out the important indexes of corporate social responsibility (CSR) from

Table 6: Sample answer in pairwise comparison (based on the dimension of economic).

	Absolutely important ←				Equally important			→ Absolutely important		
Scaling	9 : 1	7 : 1	5 : 1	3 : 1	1 : 1	1 : 3	1 : 5	1 : 7	1 : 9	Scaling
E1				V						E2
E1		V								E3
E1			V							E4
E1						V				E5
E2			V							E3
E2					V					E4
E2									V	E5
E3					V					E4
E3									V	E5
E4								V		E5

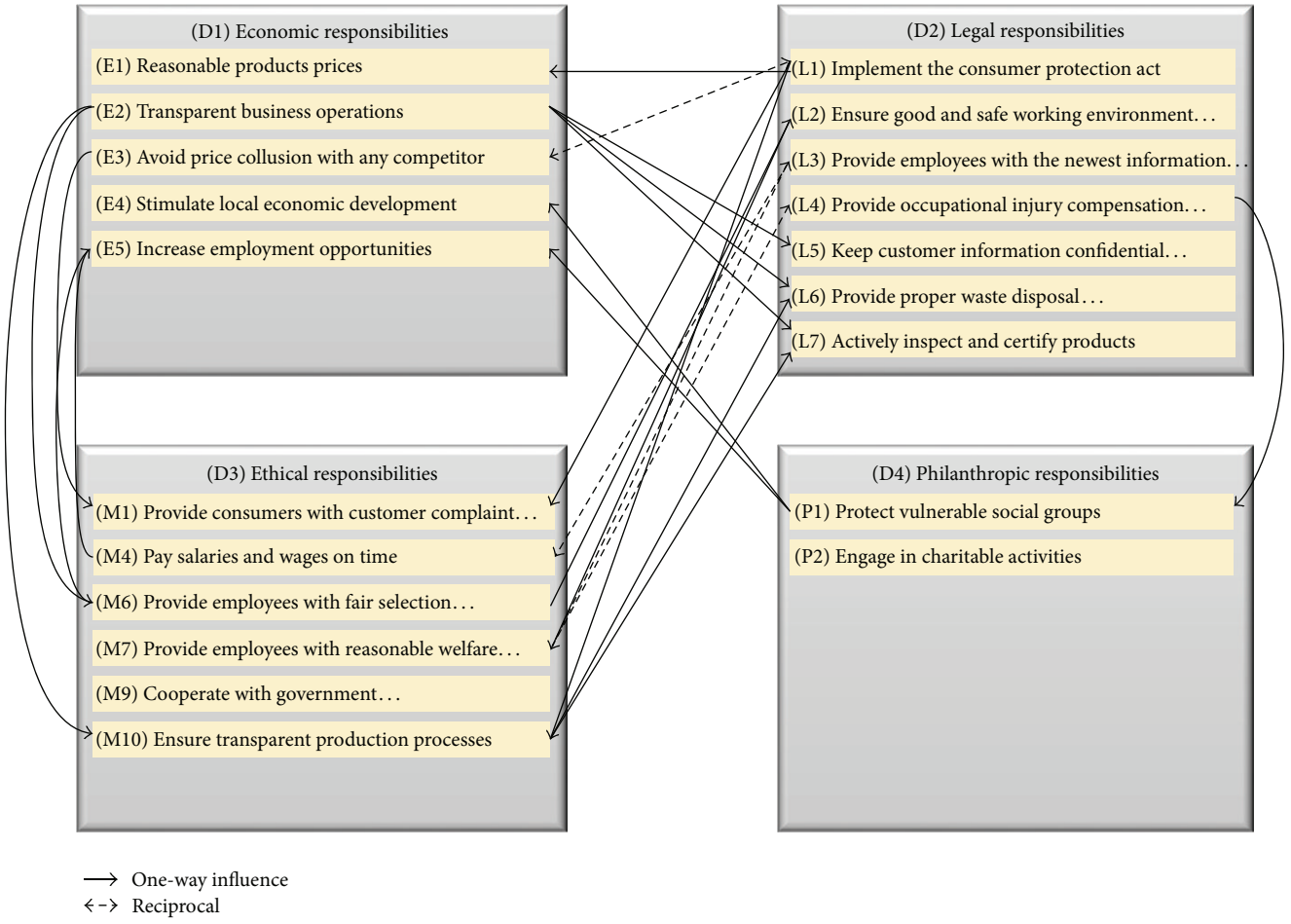


Figure 2: Network structure of the evaluation framework.

multiple perspectives. The case study results showed some interest findings. We found that "implementing the consumer protection act," "providing proper waste disposal and reduce pollutant emission," "keeping customer information confidential and protecting it against illegal use," "providing employees with the newest information of related laws," and "ensuring good and safe working environment for employees" are top five critical criteria for enterprises to implement their corporate social responsibility (CSR). Further, according to the similarity analysis for interested clusters, three groups of interest clusters can be formed, including the employees and shareholder as the first one group, the shareholders and suppliers as the second one, and the suppliers and customers as the last one group. The members of the same group

Table 7: Unweighted super matrix.

Criteria	E1	E2	E3	E4	E5	L1	L2	L3	L4	L5	L6	L7	M1	M4	M6	M7	M9	M10	P1	P2
E1	0.374	0.053	0.374	0.125	0.075	0.077	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
E2	0.304	0.043	0.391	0.217	0.043	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
E3	0.378	0.042	0.378	0.126	0.076	0.064	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
E4	0.366	0.024	0.366	0.122	0.122	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	0.000
E5	0.385	0.077	0.385	0.077	0.077	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.750	0.833	0.000	0.000	0.000	0.875	0.000
L1	0.000	0.000	**0.750**	0.000	0.000	0.026	0.179	0.128	0.179	0.026	0.231	0.231	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
L2	0.000	0.000	0.000	0.000	0.000	0.011	0.075	0.075	0.225	0.015	0.375	0.225	0.000	0.000	0.167	0.185	0.000	0.000	0.000	0.000
L3	0.000	0.000	0.000	0.000	0.000	0.011	0.057	0.057	0.285	0.019	0.285	0.285	0.000	0.250	0.000	0.659	0.000	0.000	0.000	0.000
L4	0.000	0.000	0.000	0.000	0.000	0.025	0.057	0.034	0.172	0.025	0.172	0.516	0.000	0.000	0.000	0.156	0.000	0.000	0.000	0.000
L5	0.000	0.146	0.000	0.000	0.000	0.032	0.161	0.097	0.226	0.032	0.226	0.226	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
L6	0.000	0.068	0.000	0.000	0.000	0.030	0.055	0.055	0.274	0.039	0.274	0.274	0.000	0.000	0.000	0.000	0.000	0.833	0.000	0.000
L7	0.000	0.167	0.000	0.000	0.000	0.036	0.107	0.064	0.107	0.046	0.320	0.320	0.000	0.000	0.000	0.000	0.000	0.167	0.000	0.000
M1	0.000	0.000	**0.250**	0.000	0.000	0.247	0.000	0.000	0.000	0.000	0.000	0.000	0.131	0.654	0.044	0.131	0.015	0.026	0.000	0.000
M4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.750	0.000	0.000	0.000	0.000	0.102	0.511	0.102	0.170	0.057	0.057	0.000	0.000
M6	0.000	0.586	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.198	0.330	0.066	0.330	0.009	0.066	0.000	0.000
M7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.750	0.000	0.000	0.000	0.180	0.541	0.036	0.180	0.026	0.036	0.000	0.000
M9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.265	0.265	0.206	0.206	0.029	0.029	0.000	0.000
M10	0.000	0.032	0.000	0.000	0.000	0.612	0.000	0.000	0.000	0.000	0.000	0.000	0.227	0.409	0.045	0.227	0.045	0.045	0.000	0.000
P1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.833	0.167
P2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.833	0.167

Table 8: Weighted super matrix.

Criteria	E1	E2	E3	E4	E5	L1	L2	L3	L4	L5	L6	L7	M1	M4	M6	M7	M9	M10	P1	P2
E1	0.093	0.013	0.093	0.031	0.019	0.019	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
E2	0.076	0.011	0.098	0.054	0.011	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
E3	0.095	0.011	0.095	0.032	0.019	0.016	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
E4	0.091	0.006	0.091	0.030	0.030	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.031	0.000
E5	0.096	0.019	0.096	0.019	0.019	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.188	0.208	0.000	0.000	0.000	0.219	0.000
L1	0.000	0.000	0.300	0.000	0.000	0.010	0.072	0.051	0.072	0.010	0.092	0.092	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
L2	0.000	0.000	0.000	0.000	0.000	0.004	0.030	0.030	0.090	0.006	0.150	0.090	0.000	0.000	0.067	0.074	0.000	0.000	0.000	0.000
L3	0.000	0.000	0.000	0.000	0.000	0.005	0.023	0.023	0.114	0.008	0.114	0.114	0.000	0.100	0.000	0.263	0.000	0.000	0.000	0.000
L4	0.000	0.000	0.000	0.000	0.000	0.010	0.023	0.014	0.069	0.010	0.069	0.206	0.000	0.000	0.000	0.062	0.000	0.000	0.000	0.000
L5	0.000	0.059	0.000	0.000	0.000	0.013	0.065	0.039	0.090	0.013	0.090	0.090	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
L6	0.000	0.027	0.000	0.000	0.000	0.012	0.022	0.022	0.109	0.016	0.109	0.109	0.000	0.000	0.000	0.000	0.000	0.333	0.000	0.000
L7	0.000	0.067	0.000	0.000	0.000	0.014	0.043	0.026	0.043	0.018	0.128	0.128	0.000	0.000	0.000	0.000	0.000	0.067	0.000	0.000
M1	0.000	0.000	0.050	0.000	0.000	0.049	0.000	0.000	0.000	0.000	0.000	0.000	0.026	0.131	0.009	0.026	0.003	0.005	0.000	0.000
M4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.150	0.000	0.000	0.000	0.000	0.020	0.102	0.020	0.034	0.011	0.011	0.000	0.000
M6	0.000	0.117	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.040	0.066	0.013	0.066	0.002	0.013	0.000	0.000
M7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.050	0.150	0.000	0.000	0.000	0.036	0.108	0.007	0.036	0.005	0.007	0.000	0.000
M9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.053	0.053	0.041	0.041	0.006	0.006	0.000	0.000
M10	0.000	0.006	0.000	0.000	0.000	0.122	0.000	0.000	0.000	0.000	0.000	0.000	0.045	0.082	0.009	0.045	0.009	0.009	0.000	0.000
P1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.038	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	0.025
P2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	0.025

Table 9: Limit super matrix.

Criteria	E1	E2	E3	E4	E5	L1	L2	L3	L4	L5	L6	L7	M1	M4	M6	M7	M9	M10	P1	P2
E1	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037	0.037
E2	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031	0.031
E3	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036	0.036
E4	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038	0.038
E5	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060	0.060
L1	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082	0.082
L2	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068
L3	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078
L4	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061	0.061
L5	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074	0.074
L6	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109	0.109
L7	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086	0.086
M1	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033	0.033
M4	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050	0.050
M6	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028	0.028
M7	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
M9	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021	0.021
M10	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059
P1	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006
P2	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003

Table 10: Composite important weight and priority ranking of criteria.

Criterion	DEMETAL prominence score	Delphi weight	ANP weight	Composite weight	Normalization of composite weight	General ranking
E1	0.782	0.053	0.037	0.0035	0.038	13
E2	1.407	0.054	0.031	0.0040	0.043	11
E3	0.938	0.045	0.036	0.0032	0.034	14
E4	0.606	0.045	0.038	0.0027	0.029	15
E5	0.898	0.049	0.060	0.0055	0.059	9
L1	1.407	0.052	0.082	0.0102	0.109	1
L2	1.161	0.048	0.068	0.0071	0.076	5
L3	0.980	0.047	0.078	0.0072	0.077	4
L4	1.325	0.045	0.061	0.0064	0.068	7
L5	0.755	0.056	0.074	0.0073	0.078	3
L6	0.534	0.055	0.109	0.0092	0.098	2
L7	0.667	0.046	0.086	0.0066	0.070	6
M1	0.289	0.052	0.033	0.0022	0.024	17
M4	0.762	0.042	0.050	0.0037	0.040	12
M6	0.741	0.046	0.028	0.0023	0.024	16
M7	1.591	0.040	0.040	0.0041	0.044	10
M9	0.263	0.055	0.021	0.0015	0.016	18
M10	0.913	0.054	0.059	0.0061	0.065	8
P1	0.807	0.058	0.006	0.0006	0.007	19
P2	0.472	0.057	0.003	0.0002	0.003	20
Total	—	1.000	1.000	—	1.000	—

Table 11: Relative weights of alternatives under CSR criteria.

Interested parties	L1	L6	L5	L3	L2	L7	L4	M10	E5	M7	Sum of weight
A1 suppliers	0.049	0.042	0.047	0.252	0.049	0.047	0.529	0.270	0.051	0.061	1.396
A2 shareholders	0.052	0.165	0.125	0.122	0.212	0.252	0.151	0.096	0.222	0.514	1.912
A3 employees	0.147	0.217	0.214	0.036	0.537	0.529	0.047	0.046	0.563	0.210	2.547
A4 customers	0.571	0.052	0.042	0.520	0.041	0.066	0.220	0.522	0.048	0.171	2.254
A5 general public	0.181	0.523	0.572	0.070	0.162	0.106	0.053	0.065	0.116	0.043	1.891
Total	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	—

Table 12: Euclidean distance matrix for alternatives.

Interested parties	A1 suppliers	A2 shareholders	A3 employees	A4 customers	A5 general public
A1 suppliers	0.000				
A2 shareholders	0.257	0.000			
A3 employees	0.576	0.216	0.000		
A4 customers	0.258	0.423	0.726	0.000	
A5 general public	0.423	0.307	0.385	0.565	0.000

got closed viewpoint toward the CSR evaluation criteria. Therefore, to maximize the CSR policies effect, the reference to the analyzed results can help the enterprises to establish and develop their CSR strategies for particular interested cluster.

The proposed HDMSM can enable a group of decision makers to implement the MCDM effectively and help them to analyze the relation and degree of mutual influence among different evaluation factors. As the case study demonstrated, the HDMSM can locate the evaluation factors with relatively

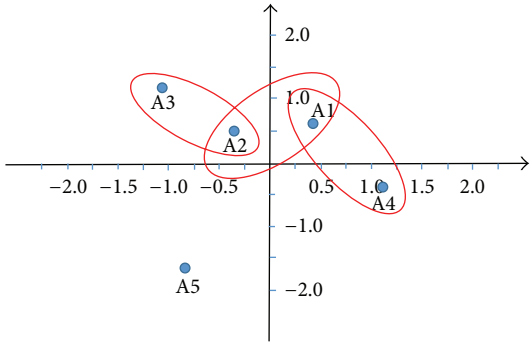


Figure 3: Perceptual map of alternative allocation.

significant and deep influence and conduct a cross analysis for different alternatives of the decision problem. Therefore, according to the analysis result, decision maker can choose the optimal alternative for decision problem solving. In the future, the HDMSM can be applied to various domains of decision problem, such as system introducing and enterprise resource planning. Also, the proposed HDMSM can be further combined with other decision-making analytic methods, such as association rules analysis, TOPSIS, or VIKOR, to upgrade the accuracy and effectiveness of the HDMSM in handling decision problems.

Conflict of Interests

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

References

[1] K. Yoon and C.-L. Hwang, "Manufacturing plant location analysis by multiple attribute decision making: part I—single-plant strategy," *International Journal of Production Research*, vol. 23, no. 2, pp. 345–359, 1985.

[2] H. A. Simon, *The New Science of Management Decision*, Harper & Row, New York, NY, USA, 1960.

[3] R. L. Keeney and H. Raiffa, *Decision with Multiple Objectives: Preference and Value Tradeoffs*, John Wiley & Sons, New York, NY, USA, 1976.

[4] Y. H. Hung, S. C. T. Chou, and G. H. Tzeng, "Knowledge management adoption and assessment for SMEs by a novel MCDM approach," *Decision Support Systems*, vol. 51, no. 2, pp. 270–291, 2011.

[5] C.-H. Hsu, F.-K. Wang, and G.-H. Tzeng, "The best vendor selection for conducting the recycled material based on a hybrid MCDM model combining DANP with VIKOR," *Resources, Conservation and Recycling*, vol. 66, pp. 95–111, 2012.

[6] W.-Y. Chiu, G.-H. Tzeng, and H.-L. Li, "A new hybrid MCDM model combining DANP with VIKOR to improve e-store business," *Knowledge-Based Systems*, vol. 37, pp. 48–61, 2013.

[7] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Application*, Springer, New York, NY, USA, 1981.

[8] R. H. Sprague Jr., "A framework for the development of decision support systems," *MIS Quarterly: Management Information Systems*, vol. 4, no. 4, pp. 1–26, 1980.

[9] C. Koo, T. Hong, and J. Kim, "A decision support system for determining the optimal size of a new expressway service area: focused on the profitability," *Decision Support Systems*, vol. 67, pp. 9–20, 2014.

[10] J. Gottschlich and O. Hinz, "A decision support system for stock investment recommendations using collective wisdom," *Decision Support Systems*, vol. 59, no. 1, pp. 52–62, 2014.

[11] Z.-H. Hu and Z.-H. Sheng, "A decision support system for public logistics information service management and optimization," *Decision Support Systems*, vol. 59, no. 1, pp. 219–229, 2014.

[12] J. A. Alalwan, "A taxonomy for decision support capabilities of enterprise content management systems," *Journal of High Technology Management Research*, vol. 24, no. 1, pp. 10–17, 2013.

[13] S. Damart, L. C. Dias, and V. Mousseau, "Supporting groups in sorting decisions: methodology and use of a multi-criteria aggregation/disaggregation DSS," *Decision Support Systems*, vol. 43, no. 4, pp. 1464–1475, 2007.

[14] E. Turban, D. Zhou, and J. Ma, "A group decision support approach to evaluating journals," *Information & Management*, vol. 42, no. 1, pp. 31–44, 2004.

[15] J. Liebenau and G. Harindranath, "Organizational reconciliation and its implications for organizational decision support systems: a semiotic approach," *Decision Support Systems*, vol. 33, no. 4, pp. 389–398, 2002.

[16] M. R. Tanniru and H. K. Jain, "Knowledge-based GDSS to support reciprocally interdependent decisions," *Decision Support Systems*, vol. 5, no. 3, pp. 287–301, 1989.

[17] G. DeSanctis and R. B. Gallupe, "A foundation for the study of group decision support systems," *Management Science*, vol. 33, no. 5, pp. 589–609, 1987.

[18] G. P. Huber, "Issues in the design of group decision support sytems," *MIS Quarterly: Management Information Systems*, vol. 8, no. 3, pp. 195–204, 1984.

[19] Y. P. Yang, H. M. Shieh, J. D. Leu, and G. H. Tzeng, "A novel hybrid MCDM model combined with DEMATEL and ANP with applications," *International Journal of Operations Research*, vol. 5, no. 3, pp. 160–168, 2008.

[20] H. A. Linstone and M. Turoff, *The Delphi Method: Techniques and Applications*, Addison-Wesley, Reading, Mass, USA, 1975.

[21] L. C. Yang and H. P. Lu, "Applying multiple criteria decision method (MCDM) to evaluate the key factors of a knowledge management system," *Information Engineering Letters*, vol. 2, no. 3, pp. 28–34, 2012.

[22] E. Fontela and A. Gabus, *Current Perceptions of the World Problematique*, North Holland, New York, NY, USA, 1976.

[23] T. L. Saaty, *Decision Making with Dependence and Feedback: The Analytic Network Process*, RWS Publications, Pittsburgh, Pa, USA, 2001.

[24] W. S. Torgerson, "Multidimensional scaling: I. Theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[25] A. B. Carroll, "Corporate social responsibility: evolution of a definitional construct," *Business and Society*, vol. 38, no. 3, pp. 268–295, 1999.

[26] A. B. Carroll, "Social issues in management research: experts' views, analysis, and commentary," *Business & Society*, vol. 33, no. 1, pp. 5–29, 1994.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 925765, 9 pages
http://dx.doi.org/10.1155/2015/925765

Research Article

A Risk-Averse Inventory Model with Markovian Purchasing Costs

Sungyong Choi[1] and Kyungbae Park[2]

[1]*Division of Business Administration, College of Government and Business, Yonsei University, 1 Yonseidae-gil, Wonju, Gangwon-do 220-710, Republic of Korea*
[2]*Department of Business Administration, College of Economics and Business Administration, Sangji University, 83 Sangjidae-gil, Wonju, Gangwon-do 220-702, Republic of Korea*

Correspondence should be addressed to Kyungbae Park; kbpark@sangji.ac.kr

Received 10 October 2014; Accepted 25 June 2015

Academic Editor: Haipeng Peng

We study a few dynamic risk-averse inventory models using additive utility functions. We add Markovian behavior of purchasing costs in our models. Such Markovian purchasing costs can reflect a market situation in a global supply chain such as fluctuations at exchange rates or the existence of product spot markets. We provide our problem formulations with finite and infinite MDP (Markovian Decision Process) problems. For finite time models, we first prove (joint) concavity of the model for each state and obtain a (modified) base-stock optimal policy. Then, we conduct comparative static analysis for model parameters and derive monotone properties to the optimal solutions. For infinite time models, we show the existence of stationary base-stock optimal policies and the inheritance of the monotone properties proven at our finite time models.

1. Introduction

Various inventory problems have been studied as dynamic programming models in supply chain literature. In these models, there exists a product to be sold over a (finite or infinite) time horizon. On the one hand, when demand exceeds supply for the product, the shortage amount is backordered and fulfilled by the supply in the next period with a backordering cost. On the other hand, when supply exceeds demands for the product, the excessive inventory is carried for the potential demand in the next period with a holding cost. The firm's objective is to determine the optimal ordering quantity so as to optimize expected total profit or costs.

The literature focuses mainly on risk-neutral performance measures, when the firm maximizes the expected total profit or minimizes the expected total costs. It implies that inventory managers are risk-neutral. Particularly, risk neutrality provides the best decision only *on average*, so it is consistent with rational decision making. However, we cannot assume all the inventory managers are risk-neutral. In supply chain literature, Schweitzer and Cachon [1] provide an experimental evidence suggesting that inventory managers may be risk-averse for the products with high profits.

In recent years, risk-averse inventory models have received an increasing attention in literature. Most work up to date focuses on single-period (newsvendor) models. For the single-period risk-averse models, Chen et al. [2] provide an excellent review and a summary of the literature in this direction. Choi et al. [3] also review the recent literature after Chen et al. [2]. On the other hand, for the multiperiod risk-averse models, Bouakiz and Sobel [4] initiated their analysis with exponential utility functions and characterized the optimal ordering policies. After then, Chen et al. [2] and Chen and Sun [5] studied dynamic inventory models in conjunction with hedging opportunities at financial markets. From this stream of research, typical key contributions to literature are the characterization of the optimal base-stock policies in dynamic inventory models.

This paper follows after Chen et al. [2] and Chen and Sun [5] in this stream where the authors consider a few risk-averse models with finite time horizon at the former

and with infinite time horizon at the latter. Chen et al. [2] consider several joint models of inventory and pricing with and without financial hedging opportunities. They establish a consumption model with an income flow in a multiperiod model. As risk measures, they use additive exponential utility functions to analyze the models and derive state-dependent (modified) base-stock optimal policies. The work of Chen and Sun [5] is a natural extension to that of Chen et al. [2] with infinite time horizon. They also use additive exponential utility functions, but without financial hedging opportunities. As a result, they obtain state-dependent (modified) base-stock optimal policies with infinite time horizon.

In our work, we add Markovian (discrete-state) behavior of purchasing costs, which distinguishes this work from Chen et al. [2] and Chen and Sun [5] where the models parameters are fixed or at most uniquely determined deterministically by their historical trajectory. That is, they do not consider the probabilistic characteristics in their model parameters. Our Markovian purchasing costs reflect typical market situations in global supply chains such as fluctuations at exchange rates or the existence of product spot markets. By exploiting such cost changes, inventory managers can get benefits from the fluctuations. Thus, a (random) fluctuation in purchasing cost affects the optimal ordering quantity significantly, so it has been frequently studied in the literature of risk-neutral inventory models (e.g., Gavirneni [6] and Yang and Xia [7]).

Gavirneni [6] considers a risk-neutral multiperiod inventory model. By analyzing the model with both finite and infinite time horizons, he obtains a base-stock optimal policy and monotone property of the impact of fluctuations in purchasing costs on the optimal ordering quantity under some conditions. Yang and Xia [7] study a continuous-review risk-neutral inventory system with a continuous MDP (Markov Decision Process) formulation. Then, they identify some conditions where the base-stock order-up-to level is monotone by the (random) fluctuations in purchasing costs. However, both of these two works only study the corresponding risk-neutral models. After then, with Markovian purchasing costs in risk-averse models, our key contributions are to conduct a comparative static analysis with finite and infinite time horizon and obtain monotone properties to the optimal solution, which have not been studied in literature.

The remainder of this paper is organized as follows. In Section 3, we establish the models with MDP formulations using general additive utility functions. Then, in Section 3.1, we prove the propositions of the concavity of the model and state-dependent optimal base-stock policy. It implies that these propositions can be preserved with risk aversion as well as risk neutrality. In addition, for our comparative static analysis, we prove the impacts of backordering and inventory holding costs to the optimal order-up-to level. Then, with the special case of exponential utility functions at Section 3.2, we also prove the impacts of (random) price changes and cost fluctuations to the optimal solution. We then extend the analytical results to the case of infinite time horizon models at Section 4. For numerical analysis, computational results are presented to confirm the analytical results in Section 5. Finally, we provide some concluding remarks in Section 6.

2. Problem Formulation

We consider a risk-averse firm to make a sequential decision from time $t = 1, 2, \ldots, T$, where T is a length of time horizon. In each time t, it faces a nonnegative and real-bounded random demand $D_t \in [0, D_{\max}]$, where demands in different time periods are independent. It also has (linear) time-invariant resale price p, inventory holding costs h, and backordering costs per unit per period.

Let us denote x_t to be the initial on-hand inventory at time t before placing an order. Similarly, y_t is the accumulated inventory at time t after receiving an order. Lead time is given zero. So, the amount to be ordered is fulfilled instantaneously.

Let us also define fluctuations in purchasing costs. We denote K as the total number of possible values of the purchasing costs in each time t, where, without loss of generality, $c_1 \geq c_2 \geq \cdots \geq c_k \geq \cdots \geq c_K$, for $k = 1, \ldots, K$. This purchasing cost in each time t undergoes Markovian behavior with a transition matrix $r = [r_{ij}]$, where r_{ij} is the probability that the purchasing cost is c_j in the next period given that it is c_i in this period. Then, the current profit function $\overline{\Pi}_t(k, x, y)$ at time t is defined when backordering is allowed given the target on-hand inventory y, initial inventory x, and purchasing costs c_k with state k. Consider

$$\begin{aligned}\overline{\Pi}_t(k, x, y) = &-c_k(y - x) + pD_t - h(y - D_t)^+ \\ &- b(D_t - y)^+, \\ &k = 1, \ldots, K,\ t = 1, \ldots, T.\end{aligned} \tag{1}$$

In addition to profit functions, we assume that inventory managers can borrow or lend money with a risk-free interest rate r_f from financial markets. That is, we consider both consumption and profit income in our model, and the current profit $\overline{\Pi}_t$ and a (nonnegative) consumption level $f_t(\geq 0)$ at time t change the current wealth level ω_t as follows:

$$\omega_{t+1} = (1 + r_f)(\omega_t + \overline{\Pi}_t - f_t). \tag{2}$$

Equivalently, $f_t = \omega_t - \alpha\omega_{t+1} + \overline{\Pi}_t$, where $\alpha = 1/(1 + r_f)$ is a discount rate. Then, our objective function is a function of consumptions in each time such as

$$\max \mathbb{E}[U(f_1, \ldots, f_T)]. \tag{3}$$

If $U(f_1, \ldots, f_T)$ is a linear function such as $U(f_1, \ldots, f_T) = \sum_{t=1}^{T} \alpha^{t-1} f_t$, the model becomes a risk-neutral model.

The definition of additive utility model is given as

$$\max \mathbb{E}[U(f_1, \ldots, f_T)] = \sum_{t=1}^{T} u_t(f_t). \tag{4}$$

To be consistent with risk aversion, each u_t is nondecreasing and concave. As a special case of (general) additive utility function, exponential utility function has the form $u_t(f_t) = -e^{-f_t/\lambda_t}$, $\lambda_t > 0$. Here, λ_t can be translated into risk tolerance factor. Thus, lower λ_t means more risk-averse.

The original model is

$$\begin{aligned} \max_{y_t \geq x_t} \quad & \mathbb{E}\left[U\left(f_1, \ldots, f_T\right)\right] \\ \text{subject to} \quad & x_{t+1} = y_t - D_t \\ & f_t = \omega_t - \alpha\omega_{t+1} + \overline{\Pi}_t \\ & \omega_{T+1} = 0 \longleftrightarrow f_T = \omega_T + \overline{\Pi}_t. \end{aligned} \tag{5}$$

For a risk-neutral model, let $V_t(\omega, k, x)$ be profits-to-go function of the risk-neutral model up to the end of time horizon, T, when backordering is allowed given that the initial on-hand inventory is x and wealth level ω with the state k at time t. Consider

$$V_t\left(\omega, k, x\right) = \max_{y \geq x} \overline{V}_t\left(\omega, k, x, y\right), \tag{6}$$

where

$$\begin{aligned} \overline{V}_t\left(\omega, k, x, y\right) = \max_{\omega'} \Bigg\{ \mathbb{E}\Bigg[& \omega - \alpha\omega' + \overline{\Pi}_t\left(k, x, y\right) \\ & + \sum_{k_1=1}^{K} r_{kk_1} \cdot V_{t+1}\left(\omega', k_1, y - D_t\right) \Bigg] \Bigg\} \end{aligned} \tag{7}$$

with a boundary condition

$$\overline{V}_T\left(\omega, k, x, y\right) = \max_{\omega} \mathbb{E}\left[\omega + \overline{\Pi}_T\left(k, x, y\right)\right]. \tag{8}$$

Due to additivity of expected value operator, wealth and consumption levels can be separated from the model as they do not affect the optimal ordering quantity. It implies that our risk-neutral case is an income-flow model without consumption through financial markets, which is equivalent to the model in Gavirneni [6].

3. Analytical Results

3.1. Additive Utility Functions. In this subsection, we focus on additive utility function to analyze a dynamic consumption model. First, we define the value function $W_t(\omega, k, x)$ which means utility-of-profits-to-go function of additive utility up to the end of time horizon, T, when backordering is allowed given that the initial on-hand inventory is x with the purchasing costs state k and wealth level ω at time t. Consider

$$W_t\left(\omega, k, x\right) = \max_{y \geq x} \mathbb{E}\left[\overline{W}_t\left(\omega, k, x, y\right)\right], \tag{9}$$

where

$$\begin{aligned} \overline{W}_t\left(\omega, k, x, y\right) = \max_{\omega'} \Bigg\{ & U_t\left[\omega - \alpha\omega' + \overline{\Pi}_t\left(k, x, y\right)\right] \\ & + \sum_{k_1=1}^{K} r_{kk_1} \cdot W_{t+1}\left(\omega', k_1, y - D_t\right) \Bigg\} \end{aligned} \tag{10}$$

with a boundary condition

$$\overline{W}_T\left(\omega, k, x, y\right) = U_T\left(\omega + \overline{\Pi}_T\left(k, x, y\right)\right). \tag{11}$$

By an equivalent formulation $W_t(\omega - c_k x, k, x) = Y_t(\omega, k, x)$ and the modified income at time t,

$$\begin{aligned} \Pi_t\left(k, y\right) = {} & y\left(\alpha c_{k_1} - c_k\right) + D_t\left(p - \alpha c_{k_1}\right) \\ & - h\left(y - D_t\right)^+ - b\left(D_t - y\right)^+, \\ & k = 1, \ldots, K, \ t = 1, \ldots, T. \end{aligned} \tag{12}$$

Then a new problem is

$$Y_t\left(\omega, k, x\right) = \max_{y \geq x} \mathbb{E}\left[\overline{Y}_t\left(\omega, k, y\right)\right], \tag{13}$$

where

$$\begin{aligned} \overline{Y}_t\left(\omega, k, y\right) = \max_{\omega'} \Bigg\{ & U_t\left[\omega - \alpha\omega' + \Pi_t\left(k, y\right)\right] \\ & + \sum_{k_1=1}^{K} r_{kk_1} \cdot Y_{t+1}\left(\omega', k_1, y - D_t\right) \Bigg\} \end{aligned} \tag{14}$$

with a boundary condition

$$\overline{Y}_T\left(\omega, k, y\right) = U_T\left(\omega + \Pi_T\left(k, y\right)\right). \tag{15}$$

Proposition 1 (existence of a wealth dependent base-stock optimal policy). *$Y_t(\omega, k, x)$ is jointly concave in x and ω for each $k = 1, \ldots, K$. In addition, a wealth (ω) dependent base-stock policy is optimal.*

Proof. Our proof idea is induction. For $Y_T(\omega, k, x)$ at $t = T$, it is obvious to be jointly concave in x and ω, $\forall k$ as $U_T(\cdot)$ is nondecreasing and concave. Next, we assume that $Y_{t+1}(\omega, k, x)$ is jointly concave in x and ω, $\forall k$. Finally, we prove that a wealth (ω) dependent base-stock policy is optimal and $Y_t(\omega, k, x)$ is jointly concave in x and ω, $\forall k$.

First, Π_t is concave in y. Thus, $\overline{Y}_t(\omega, k, y)$ is jointly concave in y and ω, $\forall k$. It implies that $\mathbb{E}[\overline{Y}_t(\omega, k, y)]$ is jointly concave in y and ω, $\forall k$.

Then, we prove that a wealth dependent base-stock policy is optimal. Let $y^*(\omega, k)$ be an optimal solution for the problem

$$\max_{y \geq x} \mathbb{E}\left[\overline{Y}_t\left(\omega, k, y\right)\right]. \tag{16}$$

Because $\mathbb{E}[\overline{Y}_t(\omega, k, y)]$ is concave in y for given ω and k, it is optimal to order-up-to $y^*(\omega, k)$ if $x < y^*(\omega, k)$ and not to order otherwise. That is, a wealth dependent base-stock policy is optimal.

Finally, after a proper modification of Theorem A.4 *(convexity preservation under minimization)* in Porteus [8], $Y_t(\omega, k, x)$ is jointly concave in x and ω, $\forall k$. □

Now we conduct a comparative static analysis of model parameters. In fact, for single-period models, the comparative static analysis was done in Eeckhoudt et al. [9]. Then, we extend the analysis to multiperiod inventory models with general utility functions at Section 3.1 and exponential utility functions at Section 3.2. The dynamic characteristics in our multiperiod models make the analysis nontrivial and even much more challenging.

Proposition 2 (the impacts of backordering costs and inventory holding costs to the optimal base-stock level). *$\widehat{y}_t^{RA}(\omega, k, x)$ is a nondecreasing (or nonincreasing) function of b (or h) which means $Y_t(\omega, k, x)$ is supermodular (or submodular) in (b, y) (or (h, y)). That is, higher backordering costs (or inventory holding costs) mean higher (lower) order-up-to level at each time $t = 1, \ldots, T$.*

Proof. Our proof idea is to use supermodularity and has two steps. First, we find the commonality between our model and the (single-period) model in Eeckhoudt et al. [9]. Specifically, we show that our boundary case at time $t = T$ is the same as in the case of Eeckhoudt et al. [9]. Finally, we show that supermodularity preserves through time periods recursively as maximization preserves it.

First, with respect to backordering costs, our profit function definition Π_t subtracts the shortage amount multiplied by unit backordering cost b, instead of being multiplied by emergency ordering costs $\widehat{c}$ at Eeckhoudt et al. [9]. Thus, our parameter b has the same monotone property at time $t = T$ as $\widehat{c}$ in Eeckhoudt et al. [9] as our objective function $Y_T(\omega, k, y)$ does not have an iterative term. Therefore, similar to Table 1 of Eeckhoudt et al. [9], $\widehat{y}_T^{\mathrm{RA}}(\omega, k, x)$ is supermodular in (b, y).

For $t = T-1, \ldots, 1$, when Y_{t+1} is supermodular in (b, y), a convex combination of $\sum_{k_1=1}^{K} r_{kk_1} \cdot Y_{t+1}((\omega)^*, k_1, y - D)$ is also supermodular in (b, y). Then, $\Pi_t(\omega, k, y)$ is supermodular in (b, y), so is $U_t[\omega - (\omega)^* + \Pi_t(\omega, k, y)]$ because $U_t(\cdot)$ is nondecreasing and concave. Lastly, maximization preserves supermodularity by Theorem 8.2 of Porteus [8] after a proper modification.

For inventory holding costs h, we can replace it with $\overline{h} = -h$ and this new parameter $\overline{h}$ preserves supermodularity similarly done above. Thus, $Y_t(\omega, k, x)$ is submodular in (h, y). □

3.2. Additive Exponential Utility Functions. Now we use exponential utility function for further analysis in Section 3.2. To analyze it with a risk tolerance parameter λ, denote the "certainty equivalent" operator with respect to a random variable ξ to be

$$\mathscr{CE}[\xi] = -\lambda \ln \mathbb{E}\left[e^{-\xi/\lambda}\right]. \tag{17}$$

We also consider the "effective risk tolerance" per period, defined as $R_t = \sum_{\tau=t}^{T} \alpha^{\tau-t} \lambda_\tau$. It implies that $R_t = \lambda_t + \alpha R_{t+1}$.

Then, at time T with an additive exponential utility function,

$$\begin{aligned} Y_T(\omega, k, x) &= \max_{y \geq x} \mathbb{E}\left[-A_T \cdot \exp\left(-\frac{(\omega + \Pi_T(k, y))}{\lambda_T}\right)\right], \quad A_T > 0 \\ &= A_T \cdot \exp\left(-\frac{\omega}{\lambda_T}\right) \cdot \max_{y \geq x}\left\{\mathbb{E}\left[-\exp\left(\frac{-(\Pi_T(k, y))}{\lambda_T}\right)\right]\right\} \\ &= A_T \cdot \exp\left(-\frac{\omega}{\lambda_T}\right) \cdot \left\{-\exp\left(-\frac{(Z_T(k, x))}{\lambda_T}\right)\right\}, \end{aligned} \tag{18}$$

where

$$\begin{aligned} Z_T(k, x) &= \max_{y \geq x} \mathscr{CE}\left[\Pi_T(k, y)\right] \\ &= \max_{y \geq x}\left(-\lambda_T \ln \mathbb{E}\left[e^{-\Pi_T(k,y)/\lambda_T}\right]\right). \end{aligned} \tag{19}$$

Thus, at time T, the value function $Z_T(k, x)$ is equivalent to the corresponding single-period (newsvendor) problem, which is $\max_{y \geq x} \mathbb{E}[-e^{-\Pi_T(k,y)/\lambda_T}]$.

Now let us consider the next case at time t with $A_{t+1} > 0$. Assume that

$$Y_{t+1}(\omega, k, x) = -A_{t+1} \cdot \exp\left(-\frac{(Z_{t+1}(k, x) + \omega)}{R_{t+1}}\right), \tag{20}$$

$$\begin{aligned} Y_t(\omega, k, x) = \max_{y \geq x} \mathbb{E}\Bigg[\max_{\omega'}\Bigg\{-a_t &\cdot \exp\left(-\frac{\left(\omega - \alpha\omega' + \Pi_t(k, y)\right)}{\lambda_t}\right) - A_{t+1} \\ &\cdot \sum_{k_1=1}^{K} r_{kk_1} \cdot \exp\left(-\frac{\left(Z_{t+1}(k_1, y - D_t) + \omega'\right)}{R_{t+1}}\right)\Bigg\}\Bigg]. \end{aligned} \tag{21}$$

For any given (k, y), the first-order optimal condition with respect to ω' is

$$\begin{aligned} &\frac{\alpha a_t R_{t+1}}{A_{t+1}\lambda_t} \cdot \exp\left(-\frac{\left(\omega - \alpha\omega'\right)}{\lambda_t}\right) \cdot \exp\left(-\frac{\Pi_t(k, y)}{\lambda_t}\right) \\ &= \exp\left(-\frac{\omega'}{R_{t+1}}\right) \cdot \sum_{k_1=1}^{K} r_{kk_1} \cdot \exp\left(-\frac{Z_{t+1}(k_1, y - D_t)}{R_{t+1}}\right). \end{aligned} \tag{22}$$

Thus,

$$\begin{aligned}&\exp\left(\frac{\alpha\omega'}{\lambda_t}\right)\cdot\exp\left(\frac{\omega'}{R_{t+1}}\right)\\&=\frac{A_{t+1}\lambda_t}{\alpha a_t R_{t+1}}\cdot\exp\left(\frac{\omega}{\lambda_t}\right)\cdot\exp\left(\frac{\Pi_t(k,y)}{\lambda_t}\right)\\&\cdot\sum_{k_1=1}^{K}r_{kk_1}\cdot\exp\left(-\frac{Z_{t+1}(k_1,y-D_t)}{R_{t+1}}\right).\end{aligned}\tag{23}$$

After taking logarithm and calculating the equation, the maximizer $(\omega')^*$ of ω' is as follows:

$$\begin{aligned}\left(\omega'\right)^*&=\left(\frac{R_{t+1}}{R_t}\right)(\omega+\Pi_t(k,y))+\left(\frac{\lambda_t R_{t+1}}{R_t}\right)\\&\cdot\ln\left(\frac{A_{t+1}\lambda_t}{\alpha a_t R_{t+1}}\right.\\&\left.\cdot\sum_{k_1=1}^{K}r_{kk_1}\cdot\exp\left(-\frac{Z_{t+1}(k_1,y-D_t)}{R_{t+1}}\right)\right).\end{aligned}\tag{24}$$

Then, the optimal consumption level f_t^* can be calculated as follows:

$$\begin{aligned}f_t^*&=\omega-\alpha\cdot\left(\omega'\right)^*+\Pi_t(k,y)\\&=-\left(\frac{\alpha\lambda_t R_{t+1}}{R_t}\right)\ln\left(\frac{A_{t+1}\lambda_t}{\alpha a_t R_{t+1}}\right)+\left(\frac{\lambda_t}{R_t}\right)\left(\omega\right.\\&\quad+\Pi_t(k,y)-\alpha R_{t+1}\\&\left.\quad\cdot\sum_{k_1=1}^{K}r_{kk_1}\cdot\exp\left(-\frac{Z_{t+1}(k_1,y-D_t)}{R_{t+1}}\right)\right)\\&=C_t+B_t\left(\omega+\Pi_t(k,y)+\alpha H_{t+1}(k,y-D_t)\right),\end{aligned}\tag{25}$$

where

$$\begin{aligned}C_t&=-\left(\frac{\alpha\lambda_t R_{t+1}}{R_t}\right)\ln\left(\frac{A_{t+1}\lambda_t}{\alpha a_t R_{t+1}}\right)\\B_t&=\frac{\lambda_t}{R_t}\in(0,1)\\H_{t+1}(k,y-D_t)&=-R_{t+1}\\&\quad\cdot\ln\left(\sum_{k_1=1}^{K}r_{kk_1}\cdot\exp\left(-\frac{Z_{t+1}(k_1,y-D_t)}{\lambda_{t+1}}\right)\right).\end{aligned}\tag{26}$$

Then, by plugging (22) into (21), it is

$$\begin{aligned}Y_t(\omega,k,x)&=\max_{y\geq x}\mathbb{E}\left[-a_t\exp\left(-\frac{f_t^*}{\lambda_t}\right)-\frac{\alpha a_t R_{t+1}}{\lambda_t}\exp\left(-\frac{f_t^*}{\lambda_t}\right)\right]\\&=\left(\frac{a_t}{B_t}\right)\max_{y\geq x}\mathbb{E}\left[-\exp\left(-\frac{f_t^*}{\lambda_t}\right)\right]\\&=\left(\frac{a_t}{B_t}\right)\max_{y\geq x}\mathbb{E}\left[-\exp\left(-\frac{(C_t+B_t(\omega+\Pi_t(k,y)+\alpha H_{t+1}(k,y-D_t)))}{\lambda_t}\right)\right]\\&=\left(\frac{a_t}{B_t}\right)\exp\left(-\frac{(C_t+B_t\omega)}{\lambda_t}\right)\max_{y\geq x}\left\{-\mathbb{E}\left[\exp\left(-\frac{(B_t(\Pi_t(k,y)+\alpha H_{t+1}(k,y-D_t)))}{\lambda_t}\right)\right]\right\}\\&=\left(\frac{a_t}{B_t}\right)\exp\left(-\frac{(C_t+B_t\omega)}{\lambda_t}\right)\max_{y\geq x}\left\{-\mathbb{E}\left[\exp\left(-\frac{(\Pi_t(k,y)+\alpha H_{t+1}(k,y-D_t))}{R_t}\right)\right]\right\}\\&=\left(\frac{a_t}{B_t}\right)\exp\left(-\frac{(C_t+B_t\omega)}{\lambda_t}\right)\left(-\exp\left(-\frac{Z_t(k,x)}{R_t}\right)\right),\end{aligned}\tag{27}$$

where

$$\begin{aligned}Z_t(k,x)&=\max_{y\geq x}\left(-R_t\right.\\&\left.\cdot\ln\mathbb{E}\left[\exp\left(-\frac{(\Pi_t(k,y)+\alpha H_{t+1}(k,y-D_t))}{R_t}\right)\right]\right)\\&=\max_{y\geq x}\left(\mathscr{C}\mathscr{C}\left(\Pi_t(k,y)+\alpha H_{t+1}(k,y-D_t)\right)\right).\end{aligned}\tag{28}$$

Therefore, the optimal order-up-to level is independent of wealth level with exponential utility function which simplifies the model.

Proposition 3 (the impact of resale price to the optimal base-stock level). *$\widehat{y}_t^{RA}(k,x)$ is a nonincreasing function of p which means $Z_t(k,x)$ is submodular in (p,y). That is, higher*

resale price means lower order-up-to level at each time $t = 1, \ldots, T$.

Proof. We will prove this proposition similar to Proposition 2.

First, our profit function definition has the same revenue part pD_t as in Eeckhoudt et al. [9]. Thus, our parameter p has the same monotone property at time $t = T$ as in Eeckhoudt et al. [9]. Therefore, similar to Table 1 of Eeckhoudt et al. [9], $\widehat{y}_T^{\mathrm{RA}}(k, x)$ is submodular in (p, y), equivalently supermodular in $(\overline{p}, y)$ with $\overline{p} = -p$.

Next, in order to discuss preservation of supermodularity for

$$\begin{aligned}\overline{\overline{H}}_{t+1}(k, Z_{t+1}) &= -R_{t+1} \\ &\quad \cdot \ln\left(\sum_{k_1=1}^{K} r_{kk_1} \cdot \exp\left(-\frac{Z_{t+1}(k_1, y-D)}{\lambda}\right)\right)\end{aligned} \tag{29}$$

in $(\overline{p}, y)$, let me work it stepwise. For $t = T-1, \ldots, 1$, when Z_{t+1} is supermodular in $(\overline{p}, y)$, $\exp(-Z_{t+1})$ is submodular in $(\overline{p}, y)$. Then, this submodularity is invariant to a convex combination of $r_{k1}, \ldots, r_{kK}$. Next, submodularity changes to supermodularity by $-\ln(\cdot)$. Finally, $\Pi_t(k, y) + \alpha H_{t+1}(k, y-D)$ is supermodular in $(\overline{p}, y)$ as $\Pi_t(k, y)$ is also supermodular. Lastly, certainty equivalent operator and maximization preserve supermodularity. □

Proposition 4 (the impact of fluctuations to the optimal base-stock level). *When* $r_{kk_1} = \tau_{k_1}$ *for each* $k = 1, \ldots, K$ *(the cost in a period is independent of the cost in the previous period), the base-stock solution is order-preserving with respect to the costs such as* $\widehat{y}_t^{RA}(k+1, x) \geq \widehat{y}_t^{RA}(k, x)$, *for all* $t = 1, \ldots, T$.

Proof. Our proof idea is to use the concept of supermodularity. First, let me denote $\overline{Z}_t(k, y)$ as follows:

$$\overline{Z}_t(k, y) = \begin{cases} \mathscr{CE}\left[\Pi_t(k, y) + \alpha H_{t+1}(k, y - D_t)\right], & t \neq T \\ \mathscr{CE}\left[\Pi_T(k, y)\right], & t = T. \end{cases} \tag{30}$$

For supermodularity of $Z_t(k, x)$ with respect to (k, y), it is equivalent to prove that

$$\frac{\partial \overline{Z}_t(k+1, y)}{\partial y} - \frac{\partial \overline{Z}_t(k, y)}{\partial y} \geq 0. \tag{31}$$

In this proposition, it is sufficient to prove that the state space and the set of state and action spaces are lattice and that

$$\frac{\partial \overline{Z}_t(k+1, y)}{\partial y} \geq 0, \quad \text{with } y = \widehat{y}_t^{\mathrm{RA}}(k, x), \tag{32}$$

due to the concavity of $\overline{Z}_t$ proven at Proposition 1. At time $t = T$, we need to prove

$$\begin{aligned}&\mathbb{E}\left[\exp\left(-\frac{\Pi_T(k+1, y)}{\lambda_T}\right) \cdot \left(-c_{k+1} + \alpha c_{k_1} + b - (h+b) \cdot \mathbb{1}_{\{D_t < y\}}\right)\right] \\ &\geq \mathbb{E}\left[\exp\left(-\frac{\Pi_T(k+1, y)}{\lambda_T}\right) \cdot \left(-c_k + \alpha c_{k_1} + b - (h+b) \cdot \mathbb{1}_{\{D_t < y\}}\right)\right] \\ &= \mathbb{E}\left[\exp\left(-\frac{\Pi_T(k, y)}{\lambda_T} - \frac{(c_k - c_{k+1})\, y}{\lambda_T}\right) \cdot \left(-c_k + \alpha c_{k_1} + b - (h+b) \cdot \mathbb{1}_{\{D_t < y\}}\right)\right] \\ &= \exp\left(-\frac{(c_k - c_{k+1})\, y}{\lambda_T}\right) \mathbb{E}\left[\exp\left(-\frac{\Pi_T(k, y)}{\lambda_T}\right) \cdot \left(-c_k + \alpha c_{k_1} + b - (h+b) \cdot \mathbb{1}_{\{D_t < y\}}\right)\right] = 0,\end{aligned} \tag{33}$$

where $y = \widehat{y}_T^{\mathrm{RA}}(k, x)$. The first inequality holds true as $c_k \geq c_{k+1}$ and $\exp(-\Pi_T(k+1, y)/\lambda_T) \geq 0$. Finally, the state space $\mathcal{S} = \{(1, \ldots, K), [-D_{\max} \times T, D_{\max} \times T]\}$ and action space $[x, D_{\max} \times T]$ are lattices trivially to satisfy the definition of lattice.

For time $t = T-1, T-2, \ldots, 1$, what we need to prove is that $\Pi_t(k, y) + \alpha H_{t+1}(k, y-D)$ is supermodular in (k, y) because certainty equivalent operator (refer to Table 1 of Topkis [10]) and maximization, which was discussed at the case of $t = T$ previously, preserve supermodularity. As $\Pi_t(k, y)$ is supermodular in (k, y), a sufficient condition to the supermodularity of $\Pi_t(k, y) + \alpha H_{t+1}(k, y-D)$ in (k, y) is that $H_{t+1}(k, y-D)$ is also supermodular in (k, y). Then, when $r_{kk_1} = \tau_{k_1}$ for all $k = 1, \ldots, K$, $H_{t+1}(k, y-D) \equiv \overline{H}_{t+1}(y-D)$ is not a function of k, and thus it guarantees that $\Pi_t(k, y) + \alpha H_{t+1}(k, y-D)$ is supermodular in (k, y). □

4. Extension to Infinite Time Horizon Model

In this section, we consider the infinite time horizon problem as a special and limiting case of the finite time horizon problem when $T \to \infty$. For infinite time horizon model, we focus on a subset of additive exponential utility functions at Section 3.2, denoted as $\overline{U}_t$. For this subset of $\overline{U}_t$, we specifically consider two conditions. The first is uniform boundedness of $\overline{U}_t$ where our utility, $\overline{U}_t$, has a finite value for all action and state spaces. That is, $\overline{U}_t(\cdot) < \infty$. So is

$$\left|\mathbb{E}\left[\overline{U}_t\left(\omega - \alpha\left(\omega'\right)^* + \Pi_t(k, y)\right)\right]\right| < \infty, \quad \forall(\omega, k, x, y, t). \tag{34}$$

Then, the second is $r_{kk_1} = \tau_{k_1}$ for all $k = 1, \ldots, K$ and $\lambda_t = \lambda$ for all time t to discuss similar analytical results at finite time MDP models, such as

$$Z_t(k,x) = \sup_{y \geq x} \left(\mathscr{C}\mathscr{E}\left(\Pi_t(k,y) + \alpha \overline{H}_{t+1}(y-D) \right) \right), \quad (35)$$

where

$$\begin{aligned} &\overline{H}_{t+1}(Z_{t+1}) \\ &\quad = -R_{t+1} \\ &\qquad \cdot \ln\left(\sum_{k_1=1}^{K} \tau_{k_1} \cdot \exp\left(-\frac{Z_{t+1}(k_1, y-D)}{\lambda} \right) \right) \end{aligned} \quad (36)$$

with $R_t = \lambda \sum_{\tau=0}^{t} \alpha^{t-\tau}$. Under this second condition, cost parameters are stationary.

Now it is time to consider the infinite horizon model with stationary parameters. Then, we study the model with an expected discounted profit criteria.

Proposition 5 (existence of a stationary and wealth independent base-stock optimal policy). *A stationary and wealth (ω)-independent base-stock policy is optimal with infinite time horizon model.*

Proof. There are two methods to prove stationarity of infinite time horizon model. The first one is to prove that our optimal value operator is a contraction mapping. Then, by Banach's fixed point theorem, there exists a unique optimal solution to satisfy stationarity. In this paper, we use an alternative method shown in Puterman [11]. Then, by Theorem 6.11.10 of Puterman [11], what we need to show is uniform boundedness of $\mathbb{E}[\overline{U}_t]$ which we focus on in this section as all other conditions are trivial. Then, only one difference between finite and infinite time horizon models is that maximization is replaced by supremum as we consider continuous state and action spaces. Thus, for each state, there exists

$$Z(k,x) = \lim_{T \to \infty} Z_t(k,x). \quad (37)$$

Moreover, this $Z(k,x)$ is the unique solution of

$$Z(k,x) = \sup_{y \geq x} \mathbb{E}\left[\overline{Z}(k,y) \right], \quad (38)$$

where

$$\overline{Z}(k,y) = \mathscr{C}\mathscr{E}\left[\Pi(k,y) + \alpha H(k, y-D) \right]. \quad (39)$$

□

Corollary 6 (inheritance of the monotone optimal policy for backordering costs and inventory holding costs with infinite time horizon problem). *The base-stock solution is order-preserving (or order-reversing) with respect to the backordering costs (or inventory holding costs) such that $\widehat{y}^{RA}(\omega, k, x)$ is a nondecreasing (or nonincreasing) function of b (or h).*

Proof. It is evident by Propositions 2 and 5 with Lemma 8-4 (a) of Heyman and Sobel [12] as our state space is trivially a partially ordered set. □

Corollary 7 (inheritance of the monotone policy for purchasing costs and resale price with infinite time horizon problem). *The base-stock solution is order-preserving (or order-reversing) with respect to the purchasing costs (or resale price) such as $\widehat{y}^{RA}(k+1,x) \geq \widehat{y}^{RA}(k,x)$ and $\widehat{y}^{RA}(k,x)$ is a nonincreasing function of p.*

Proof. The proof is the same as in Corollary 6. □

5. Computational Study

In this section, we provide our numerical results to confirm the analytical results in Section 3. We consider additive exponential utility with $\lambda_t = \lambda$, for all $t = 1, \ldots, T$. For model parameters, we highlight a planning horizon with $T = 10$. Then, as a base case, we set up a resale price, $p = 400$, backordering costs, $b = 150$, and inventory holding costs, $h = 10$, with a discount rate, $\alpha = 1/(1+r_f) = 0.9$. In addition, we also define Markovian fluctuation of purchasing costs, $c = [80, 65, 50, 35, 20]$ with $K = 5$. Our transition matrix is $r = [r_{ij}] = [\tau_j]$ with $\tau_j = [0.1, 0.2, 0.4, 0.2, 0.1]$ for all $j = 1, \ldots, K$. That is, a purchasing cost in a period is independent of the cost in the previous period. Finally, demands in each time are *iid* and have a support $[0, D_{\max}]$ with $D_{\max} = 100$. Then, it has a truncated and discretized normal distribution bounded by 0 and $D_{\max}$ where the expected value and variance of the original (unbounded) distribution are given as $\mathbb{E}[D_t] = D_{\max}/2$ and $\mathbb{V}\mathrm{ar}[D_t] = (D_{\max}/6)^2$.

Figure 1 shows how risk tolerance factor affects the optimal solution. We select the factor to be $\lambda = 1000, 3000$ and compare the optimal order-up-to level with the risk-neutral solutions. When λ increases, the optimal solution becomes higher and eventually converges to the risk-neutral solution in the limit.

Figures 2–4 present the numerical results for comparative static analysis with backordering costs, inventory holding costs, and resale price, respectively. In Figure 2, we select our backordering costs to be $b = 100, 300, 600$ and all other parameters are the same as in our base case. Then, as backordering costs increase, the optimal solutions also increase for each time $t = 1, \ldots, T$ and $k = 1, \ldots, K$. For Figures 3 and 4, we study the impacts of inventory holding costs and resale price to the optimal solutions. Similarly, we take the same values as in our base case except $h = 10, 150, 300$ for Figure 3 and $p = 200, 400, 800$ for Figure 4, respectively. Then, in all our cases, our analytical results are confirmed to show monotone impacts of these model parameters such that the optimal solutions decrease when inventory holding costs or resale price increases.

6. Conclusion

This paper reconsiders risk-averse inventory models in supply chain literature. Different from the previous works in literature, we use the two key conditions simultaneously, which are multiperiod models and fluctuations in purchasing costs. Although most of the results are seemingly consistent with those in literature, they are analytically challenging and

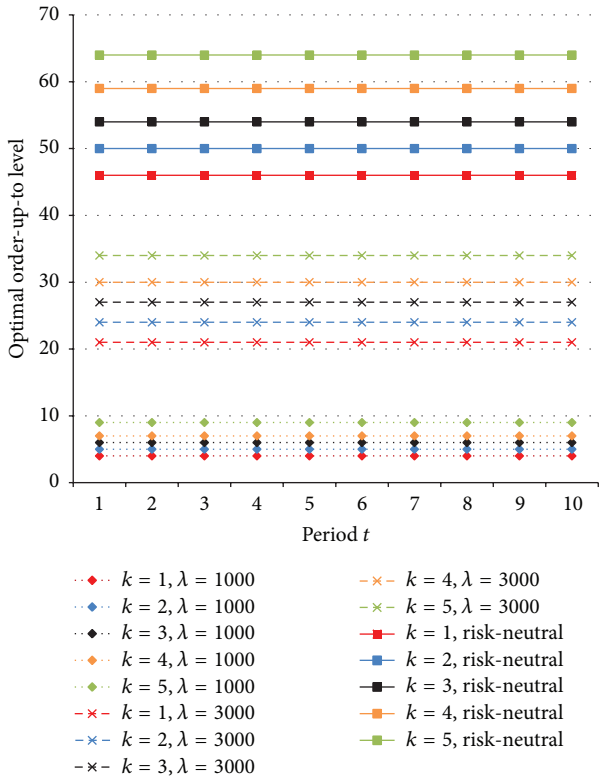


Figure 1: The impact of risk tolerance factor to optimal order-up-to level.

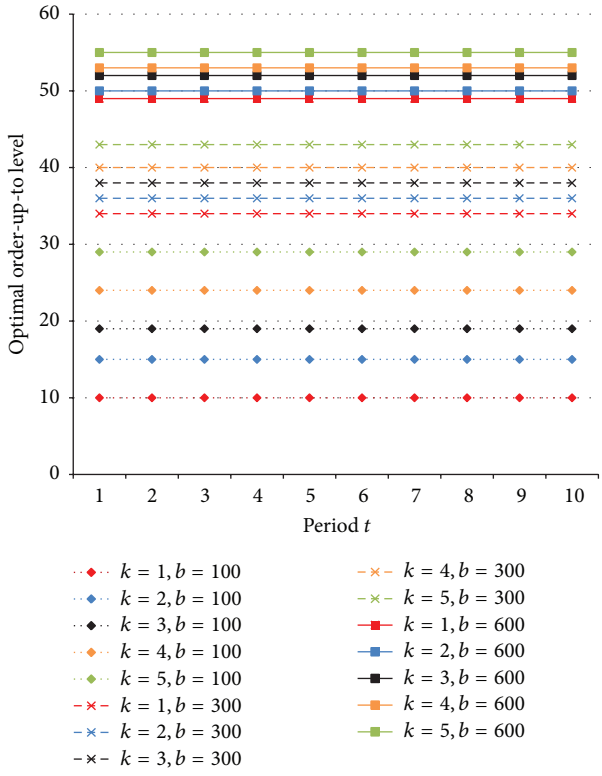


Figure 2: The impact of backordering costs to optimal order-up-to level.

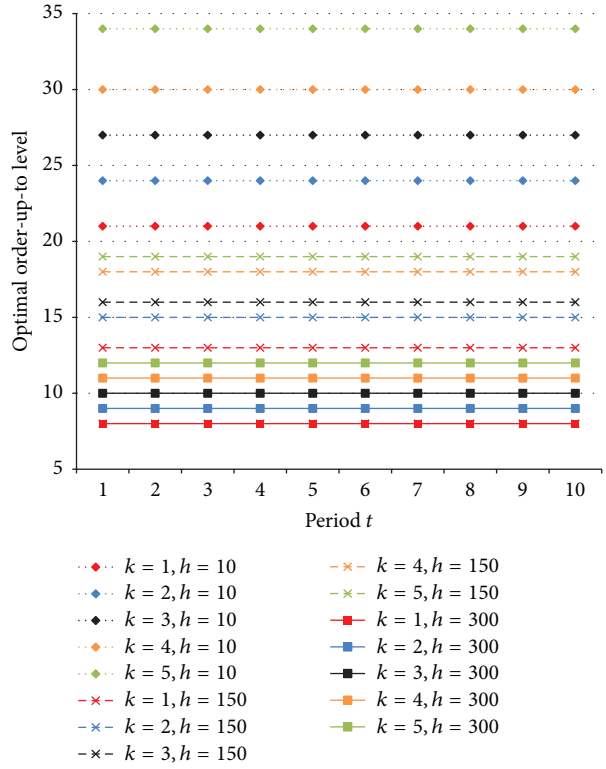


Figure 3: The impact of inventory holding costs to optimal order-up-to level.

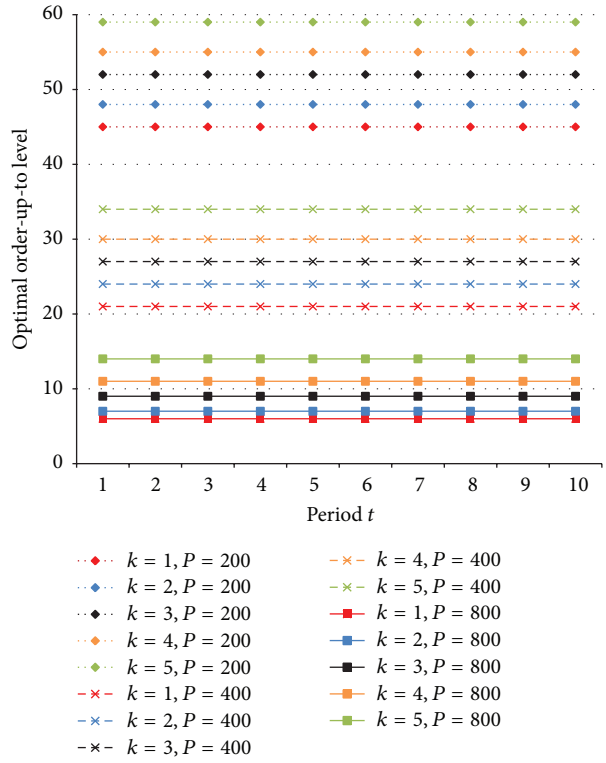


Figure 4: The impact of resale price to optimal order-up-to level.

need to be proved rigorously with independent investigation. In fact, most of the multiperiod inventory models tend to focus on characterizing the base-stock optimal ordering policies in general regardless of risk preferences. This paper could fulfill the knowledge gap in literature to conduct a comparative static analysis as a further research in this research stream.

For possible limitations, the impact of risk tolerance factor has not been discussed analytically but only numerically in this paper. Actually Figure 1 in Section 5 may imply the existence of the monotone impact on risk tolerance factor, if possible, in multiperiod inventory models. In the literature of risk-averse inventory models, such monotone impact on risk tolerance factor has been studied in various single-period models (e.g., Eeckhoudt et al. [9]). Thus, it is an interesting conjecture and would be left as a further possible line in this research stream, which has not been proved yet with any multiperiod risk-averse inventory models in literature, up to our best knowledge.

Conflict of Interests

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

Acknowledgments

The first author, Sungyong Choi, was supported (in part) by the Yonsei University Future-Leading Research Initiative of 2014. The corresponding author, Kyungbae Park, was supported by Sangji University Research Fund, 2014. The authors are very grateful to the anonymous referees for their comments and suggestions.

References

[1] M. E. Schweitzer and G. P. Cachon, "Decision bias in the newsvendor problem with a known demand distribution: experimental evidence," *Management Science*, vol. 46, no. 3, pp. 404–420, 2000.

[2] X. Chen, M. Sim, D. Simchi-Levi, and P. Sun, "Risk aversion in inventory management," *Operations Research*, vol. 55, no. 5, pp. 828–842, 2007.

[3] S. Choi, A. Ruszczyński, and Y. Zhao, "A multiproduct risk-averse newsvendor with law-invariant coherent measures of risk," *Operations Research*, vol. 59, no. 2, pp. 346–364, 2011.

[4] M. Bouakiz and M. J. Sobel, "Inventory control with an exponential utility criterion," *Operations Research*, vol. 40, no. 3, pp. 603–608, 1992.

[5] X. Chen and P. Sun, "Optimal structural policies for ambiguity and risk averse inventory and pricing models," *SIAM Journal on Control and Optimization*, vol. 50, no. 1, pp. 133–146, 2012.

[6] S. Gavirneni, "Periodic review inventory control with fluctuating purchasing costs," *Operations Research Letters*, vol. 32, no. 4, pp. 374–379, 2004.

[7] J. Yang and Y. Xia, "Acquisition management under fluctuating raw material prices," *Production and Operations Management*, vol. 18, no. 2, pp. 212–225, 2009.

[8] E. L. Porteus, *Foundations of Stochastic Inventory Theory*, Stanford University Press, Stanford, Calif, USA, 2002.

[9] L. Eeckhoudt, C. Gollier, and H. Schlesinger, "The risk-averse (and prudent) newsboy," *Management Science*, vol. 41, no. 5, pp. 786–794, 1995.

[10] D. M. Topkis, "Minimizing a submodular function on a lattice," *Operations Research*, vol. 26, no. 2, pp. 305–321, 1978.

[11] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley-Interscience, New York, NY, USA, 2nd edition, 1994.

[12] D. P. Heyman and M. J. Sobel, *Stochastic Models in Operations Research: Volume II, Stochastic Optimization*, Dover Publications, Mineola, NY, USA, 2004.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 501915, 16 pages
http://dx.doi.org/10.1155/2015/501915

Research Article

Maintaining Track Continuity for Extended Targets Using Gaussian-Mixture Probability Hypothesis Density Filter

Yulan Han, Hongyan Zhu, and ChongZhao Han

MOE KLINNS Laboratory, Institute of Integrated Automation, School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, Shanxi 710049, China

Correspondence should be addressed to Hongyan Zhu; hyzhu2011@gmail.com

Received 26 September 2014; Revised 2 February 2015; Accepted 3 February 2015

Academic Editor: Dan Simon

A multiextended-target tracker based on the extended target Gaussian-mixture probability hypothesis density (ET-GMPHD) filter, which can provide the tracks of the extended targets, is proposed to maintain the track continuity for the extended targets. To identify the extended targets, each individual Gaussian term of the mixture representing the posterior intensity function will be assigned a label, which is evolved through time. Then a track management scheme, including track initiation, track confirmation, track propagation, and termination, is developed to form the tracks for the extended targets. Furthermore, to improve the performance of the extended target tracker we also propose a mixture partitioning algorithm for resolving the identities of the extended targets in close proximity. The simulation results show that our proposed tracker achieves the less error of the position estimates and decreases the probability of incorrect label assignments from 0.6 to 0.25.

1. Introduction

In general target tracking applications, it is assumed that each target produces at most one measurement per time step. This is true when their extension is assumed to be negligible in comparison with sensor resolution. However, with the increasing sensor resolution capability this assumption is no longer valid. For instance, in short-range and maritime surveillance applications, different scattering centers, which vary from scan to scan both in its number and the relative origin location, of the objects may give rise to several distinct detections.

Extended target tracking has attracted much attention in the last decade. Gilholm and Salmond in [1] presented an approach for tracking extended targets under the assumption that the number of received target measurements in each time step is Poisson distributed. An inhomogeneous Poisson point process measurement model was developed in [2]. This measurement model could imply that the extended target is sufficiently far away from the sensor for its measurements to resemble a cluster of points, rather than a geometrically structured ensemble. A similar approach was proposed in [3] where track-before-detect theory was applied to track a 1D extent target. Baum et al. presented a random hypersurface model, which was used to track elliptic targets in [4, 5] and more general shapes in [6]. Another method to elliptic target model is an approximate Bayesian approach based on the random matrix by Koch in [7]. The target kinematical states are modeled by a Gaussian distribution, and the ellipsoidal target extension is modeled by a random matrix which follows the inverse Wishart distribution. In [8], Koch and Feldmann applied the filter based on the random matrix to track group targets under kinematical constraints. Modifications and improvements to the random matrix model of [7] were developed in [9]. A comparison of the random matrix model and the random hypersurface model was discussed in [10]. Other methods to obtain the target extension information were given in [11–14]. However, almost all those methods mentioned above are for single extended target.

Wieneke and Koch in [15] integrated random matrix, and Baum et al. in [16] integrated Mixture RHM into a probabilistic multihypothesis tracking (PMHT) framework to track multiply extended objects. However the complexity grows dramatically with the number of extended targets and measurements increasing. Another way to track multiple extended targets is based on the random finite set (RFS).

In [17], Mahler presented an extension of the probability hypothesis density (PHD) filter [18] to handle extended targets of the type presented in [2]. Orguner et al. proposed a cardinalized probability hypothesis density (CPHD) filter for extended targets [19]. Vo and Ma presented an extension of the Gaussian-mixture PHD filter [20] for extended targets, called the extended target GMPHD filter (ET-GMPHD) [21], and described much more details and extensive investigations of the methodology [22]. Random matrix framework was adapted into the ET-GMPHD framework by Granström and Orguner in [23], resulting in the Gaussian inverse Wishart PHD (GIW-PHD) filter. For the sake of convenience, the ET-GMPHD filter [21, 22] is referred to as the original ET-GMPHD filter from here onwards. However, the object identities were not involved in the implementations of the PHD filter such as the particle PHD filter and the GMPHD filter.

For postprocessing such as the behavior of objects and activity recognition, the track continuity of objects needs to be obtained. There are some studies on the track continuity in implementations of the PHD filter. The multiple hypothesis tracker and assignment algorithms are applied to the particle PHD filter to form the tracks of targets in [24] and [25, 26], respectively. Moreover, there are some methods which analyze the propagation of particles to maintain the track continuity [27, 28]. Due to the unreliability of the clustering methods in the particle PHD filter, the performance of these approaches may be affected. Recently, Clark et al. introduced a technique to identify the state estimates of objects in the GMPHD filter [29]. This method was successfully applied in sonar image tracking [30]. However, the temporal information which adversely affects the performance was not included. Pham et al. [31] proposed a method for maintaining the continuity of state estimates of objects in the GMPHD filter. The set of labels from Gaussian components was used to create hypotheses for label association process and the Hungarian algorithm was applied to search for the best hypothesis association. Panta et al. [32] proposed a GMPHD filter-based multitarget tracker, which provided track labels and the association amongst state estimates of targets over time. Various issues regarding initiating, propagating, and terminating tracks were discussed. However, to the best of our knowledge, no work was carried out on the track continuity in the ET-GMPHD filter.

In the works based on RFS discussed above, theoretically all the possible partitions of the measurements should be considered. However, the number of all the possible partitions grows dramatically with the increase in the number of measurements. Distance partitioning (DP) and distance partitioning with subpartitioning (DP-SP) were proposed to obtain a reasonable subset of all the possible partitions in [22]. In [33], to further reduce the computation time, Zhang and Ji proposed a ART (adaptive resonance theory) partitioning algorithm based on the fuzzy ART. Moreover, subpartitioning was applied to handle spatially close targets. Scheel proposed a mixture clustering algorithm, which could decompose the GIW-PHD filtering procedure into independent problems and thus reduce the combinatorial and computational complexity significantly [34]. By using the mixture cluster method, the GIW-PHD filter could be applied to the real-time tracking applications. In [33, 34], the problem with underestimation of the target number when there are extended targets in close proximity was not discussed further.

In this paper, we propose a multiextended-target tracker based on the original ET-GMPHD filter [21, 22], which provides not only the state estimates of targets at each time step but also the association of state estimates to targets over time. Three main contributions have been made to achieve this purpose, just as follows.

(i) To obtain the temporal association for the state estimates of individual extended targets, we assign the labels to individual Gaussian terms and develop a method of the label evolution through time. State trajectories of the individual extended targets can be obtained directly from the evolution of the Gaussian mixture.

(ii) We propose a track management scheme of initiating, confirming, propagating, and terminating tracks to construct the tracks of the extended targets.

(iii) To reduce the label assignment error when there are spatially close extended targets, mixture partitioning algorithm is proposed.

The rest of the paper is organized as follows. We briefly describe the extended target tracking problem in Section 2. Section 3 provides a summary of the original ET-GMPHD filter [22]. Section 4 presents the ET-GMPHD tracker proposed in this paper. In Section 5, simulation results are given to demonstrate the performance of the proposed ET-GMPHD tracker. Section 6 draws the conclusion and outlines future research directions.

2. Problem Formulation

The aim of the ET-GMPHD filter is to estimate the set of the extended target states $X_k = \{\mathbf{x}_k^{(i)}\}_{i=1}^{N_{x,k}}$, given the sets of measurements $\mathbf{Z}_k = \{\mathbf{z}_k^{(i)}\}_{i=1}^{N_{z,k}}$, for discrete time instants $k = 1, \ldots, K$, where $N_{x,k}$ is the unknown number of targets and $N_{z,k}$ is the number of measurements. The purpose of the ET-GMPHD tracker is to provide the track set $\mathcal{T}_k = \{T_k^{(i)}\}_{i=1}^{N_k^T}$, where N_k^T is the number of tracks in track set $\mathcal{T}_k$. Here, each track $T_k^{(i)}$ contains the set of estimated target states $\mathbf{X}_k^{(i)}$, corresponding time steps set $t_k^{(i)}$ and its label $l_k^{(i)}$. $\mathbf{X}_k^{(i)}$ contains all the estimated states of the extended target, whose label equals $l_k^{(i)}$, and from the time step it appears to the time step k.

The dynamic evolution of each target state $\mathbf{x}_k^{(i)}$ is assumed to be modeled by a linear Gaussian dynamic model

$$\mathbf{x}_{k+1}^{(i)} = \mathbf{F}_k \mathbf{x}_k^{(i)} + \mathbf{G}_k \mathbf{w}_k^{(i)} \tag{1}$$

for $i = 1, \ldots, N_{x,k}$, where $\mathbf{w}_k^{(i)}$ is Gaussian white noise with the covariance $\mathbf{Q}_k^{(i)}$. It is assumed that each target state evolves according to the same dynamic model independent of the other targets.

The measurements originating from the ith target are assumed to be related to the target state according to a linear Gaussian model

$$\mathbf{z}_k^{(j)} = \mathbf{H}_k \mathbf{x}_k^{(i)} + \mathbf{G}_k \mathbf{v}_k^{(j)}, \tag{2}$$

where $\mathbf{v}_k^{(j)}$ is white Gaussian noise with covariance $\mathbf{R}_k$. Each target is assumed to give rise to measurements independently of the other targets. We emphasize here that in an RFS framework both the set of measurements $\mathbf{Z}_k$ and the set of target states X_k are unlabeled, and hence no assumptions are made regarding which target gives rise to which measurement. The number of measurements generated by the ith target at each time step, denoted by $N_{m,k}^{(i)}$, is a Poisson distributed random variable with rate $\gamma(\mathbf{x}_k^{(i)})$ measurements per scan.

At each time step, clutter measurements are also generated. The number of clutter measurements N_k^c is a Poisson distributed random variable with rate $\beta_{FA,k}$ clutter measurements per surveillance volume per scan. The clutter measurements uniform over the surveillance volume.

3. Review of the Original ET-GMPHD Filter

The original ET-GMPHD filter [21, 22] is reviewed in Section 3.1. Section 3.2 describes two methods of partitioning the measurement set.

3.1. The Original ET-GMPHD Filter. Since the Gaussian mixture prediction equations of the ET-GMPHD filter are the same as those of the standard GMPHD filter [20], only measurement update formulas of the ET-GMPHD filter are introduced below. Here, six assumptions which are made in [20] hold here.

In the standard GMPHD-filter measurement update, each measurement is used to update each Gaussian component. In the ET-GMPHD filter, each cell of each partition is applied to update each Gaussian component. The corrected PHD-intensity, which is derived in [17], is the multiplication of the predicted PHD and the measurement pseudolikelihood function L_{Z_k},

$$v_{k|k}(\mathbf{x} \mid Z) = L_{Z_k}(\mathbf{x})\, v_{k|k-1}(\mathbf{x} \mid Z). \tag{3}$$

The measurement pseudolikelihood function L_{Z_k} is defined as

$$L_{Z_k}(\mathbf{x}) = 1 - \left(1 - e^{-\gamma(\mathbf{x})}\right) p_D(\mathbf{x}) + e^{-\gamma(\mathbf{x})} p_D(\mathbf{x}) \sum_{p \angle Z_k} \omega_p \sum_{W \in p} \frac{\gamma(\mathbf{x})^{|W|}}{d_W} \prod_{z \in W} \frac{\phi_{\mathbf{z}}(\mathbf{x})}{\lambda_k c_k(\mathbf{z})}, \tag{4}$$

where $p_D(\mathbf{x})$ is the detected probability of the extended target; ω_p and d_W are nonnegative coefficients defined for each partition and cell, respectively; $\phi_{\mathbf{z}}(\mathbf{x})$ is the likelihood function for a single target-generated measurement, which is a Gaussian density under the measurement model; λ_k is the mean number of clutter measurements per scan; $c_k(\mathbf{z})$ is the spatial distribution of the clutter over the surveillance region; the notation $p \angle Z_k$ means that p partitions the measurement set Z_k into nonempty cells W. The first summation is taken over all partitions p of the measurement set Z_k. The second summation is taken over all cells W in the current partition p, and the product is over all measurements in the cell W. For each partition, the measurements in cells containing more than one measurement can be interpreted as from the same target. Measurements in cells with just one measurement can be either from clutter or from target.

The first part of (4), $1 - (1 - e^{-\gamma(\mathbf{x})})\, p_D(\mathbf{x})$, handles the targets for which there are no detections. The second part handles targets for which there is at least one detection.

Assuming that the predicted intensity has the Gaussian-mixture form

$$v_{k|k-1}(\mathbf{x}) = \sum_{i=1}^{J_{k|k-1}} \omega_{k|k-1}^{(i)} \mathcal{N}\left(\mathbf{x}; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\right), \tag{5}$$

where $\mathcal{N}(\cdot; m, P)$ denotes a Gaussian density with mean m and covariance P and ω is nonnegative weight; the posterior intensity at time k is also a Gaussian mixture, as shown as follows:

$$v_{k|k}(\mathbf{x}) = v_{k|k}^{ND}(\mathbf{x}) + \sum_{p \angle Z_k} \sum_{W \in p} v_{k|k}^{D}(\mathbf{x}, W). \tag{6}$$

The Gaussian-mixture $v_{k|k}^{ND}(\cdot)$, handling the no detections case, is given by

$$\begin{aligned} v_{k|k}^{ND}(\mathbf{x}) &= \sum_{j=1}^{J_{k|k-1}} \omega_{k|k}^{(j)} \mathcal{N}\left(\mathbf{x}; m_{k|k}^{(j)}, P_{k|k}^{(j)}\right), \\ \omega_{k|k}^{(j)} &= \left(1 - \left(1 - e^{-\gamma^{(j)}}\right) p_D^{(j)}\right) \omega_{k|k-1}^{(j)}, \\ m_{k|k}^{(j)} &= m_{k|k-1}^{(j)}, \\ P_{k|k}^{(j)} &= P_{k|k-1}^{(j)}, \end{aligned} \tag{7}$$

where $\gamma^{(j)}$ and $p_D^{(j)}$ are short for $\gamma(m_{k|k-1}^{(j)})$ and $p_D(m_{k|k}^{(j)})$, respectively.

The Gaussian-mixture $v_{k|k}^{D}(\mathbf{x}, W)$, handling detected targets, is given by

$$\begin{aligned} v_{k|k}^{D}(\mathbf{x}, W) &= \sum_{j=1}^{J_{k|k-1}} \omega_{k|k}^{(j)} \mathcal{N}\left(\mathbf{x}; m_{k|k}^{(j)}, P_{k|k}^{(j)}\right), \\ \omega_{k|k}^{(j)} &= \omega_p \frac{\Gamma^{(j)} p_D^{(j)}}{d_W} \Phi_W^{(j)} \omega_{k|k-1}^{(j)}, \\ \Gamma^{(j)} &= e^{-\gamma^{(j)}} \left(\gamma^{(j)}\right)^{|W|}, \\ \Phi_W^{(j)} &= \prod_{\mathbf{z} \in W} \frac{\phi_{\mathbf{z}}\left(m_{k|k}^{(j)}\right)}{\lambda_k c_k(\mathbf{z})} \end{aligned} \tag{8}$$

and the likelihood function of one measurement is

$$\phi_{\mathbf{z}}\left(m_{k|k}^{(j)}\right) = \mathcal{N}\left(\mathbf{z} \mid \mathbf{H}_k m_{k|k-1}^{(j)}, \mathbf{R}_k + \mathbf{H}_k P_{k|k-1}^{(j)} \mathbf{H}_k^T\right). \tag{9}$$

The partition weights ω_p can be considered as the probability of the partition p being true and can be written as

$$\omega_p = \frac{\prod_{W\in p} d_W}{\sum_{p'\angle Z_k} \prod_{W'\in p'} d_{W'}},$$
$$d_W = \delta_{|W|,1} + \sum_{l=1}^{J_{k|k-1}} \Gamma^{(l)} p_D^{(l)} \Phi_W^{(l)} \omega_{k|k-1}^{(l)}, \tag{10}$$

where $\delta_{i,j}$ is the Kronecker delta. The mean and covariance of the Gaussian components are updated by using the standard Kalman measurement update,

$$m_{k|k}^{(j)} = m_{k|k-1}^{(j)} + \mathbf{K}_k^{(j)} \left(\mathbf{z}_W - \mathbf{H}_W m_{k|k-1}^{(j)}\right),$$
$$P_{k|k}^{(j)} = \left(I - \mathbf{K}_k^{(j)} \mathbf{H}_W\right) P_{k|k-1}^{(j)}, \tag{11}$$
$$\mathbf{K}_k^{(j)} = P_{k|k}^{(j)} \mathbf{H}_W^T \left(\mathbf{H}_W P_{k|k}^{(j)} \mathbf{H}_W^T + \mathbf{R}_W\right)^{-1},$$

where $\mathbf{z}_W$, $\mathbf{H}_W$, and $\mathbf{R}_W$ are defined as

$$\mathbf{z}_W \triangleq \bigoplus_{\mathbf{z}_k\in W} \mathbf{z}_k, \qquad \mathbf{H}_W = \left[\underbrace{\mathbf{H}_k^T, \mathbf{H}_k^T, \ldots, \mathbf{H}_k^T}_{|W| \text{ times}}\right]^T,$$
$$\mathbf{R}_W = \text{blkdiag}\left(\underbrace{\mathbf{R}_k, \mathbf{R}_k, \ldots, \mathbf{R}_k}_{|W| \text{ times}}\right). \tag{12}$$

The operation $\oplus$ is vertical vectorial concatenation. The number of Gaussian components increases rapidly as the time progresses. To keep the number of Gaussian components at a computationally tractable level, pruning and merging are applied as in [20].

3.2. Partitioning the Measurement Set. In (4), all the possible partitions of the measurement set are considered in an ideal situation. However, the number of all the possible partitions would grow dramatically as the size of the measurement set increases. Thus choosing a subset of all the possible partitions is necessary to achieve the acceptable computational complexity. This section describes distance partitioning and subpartitioning proposed in [22].

3.2.1. Distance Partitioning (DP). Given a set of measurements $Z = \{\mathbf{z}^{(i)}\}_{i=1}^{N_z}$ and a distance measure $\mathbf{d}(\cdot,\cdot)$, the distances between each pair of measurements can be calculated as

$$\Delta_{ij} \triangleq \mathbf{d}\left(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}\right), \quad \text{for } 1 \le i \ne j \le N_z. \tag{13}$$

It is proved in [22] that there is a unique partition that leaves all pairs (i, j) of measurements satisfying $\Delta_{ij} \le d_l$ in the same cell. N_d alternative partitions of the measurement set Z are generated by selecting N_d different thresholds

$$\{d_l\}_{l=1}^{N_d}, \quad d_l < d_{l+1}, \quad \text{for } l = 1, \ldots, N_d - 1. \tag{14}$$

For each d_l, one partition is obtained where the cells constitute sets of measurements that are no more than d_l apart from their closest cell neighbor.

The thresholds $\{d_l\}_{l=1}^{N_d}$ are selected from the set

$$\mathscr{D} \triangleq \{0\} \cup \left\{\Delta_{ij} \mid 1 \le i < j \le N_z\right\}. \tag{15}$$

If all of the elements in $\mathscr{D}$ are used to form alternative partitions, $|\mathscr{D}| = N_z(N_z - 1)/2 + 1$ partitions are achieved. To further reduce the computational load, only a subset of thresholds in the set $\mathscr{D}$ are applied to generate partitions.

The Mahalanobis distance is selected as the distance measure $\mathbf{d}(\cdot,\cdot)$. For two measurements $\mathbf{z}^{(i)}$ and $\mathbf{z}^{(j)}$ belonging to the same target, $\mathbf{d}_M(\mathbf{z}^{(i)}, \mathbf{z}^{(j)})$ is χ^2 distributed with degrees of freedom equal to the dimension of the measurement vector. A unitless distance threshold, denoted by δ_{P_G}, can be calculated as

$$\delta_{P_G} = \text{invchi2}\left(P_G\right) \tag{16}$$

for a given probability P_G, where invchi2$(\cdot)$ is the inverse cumulative χ^2 distribution function. In [21], it is illustrated that good target tracking results could be obtained in the situation that the subset of distance thresholds in $\mathscr{D}$ satisfies the condition $\delta_{P_L} < d_l < \delta_{P_U}$ with lower probabilities $P_L \le 0.3$ and upper probabilities $P_U \ge 0.8$.

3.2.2. Subpartitioning (SP). The results given by the ET-GMPHD filter with DP show the problem with underestimation of target set cardinality in situations where two or more extended targets are spatially close [21]. When targets are spatially close, so are their measurements. In this case, measurements from more than one extended target would be included in the same cell W in all partitions obtained by DP, and subsequently the ET-GMPHD filter interprets measurements from multiple targets as originating from the same target. SP was proposed in [22] to form additional partitions after performing DP.

Suppose that a set of partitions using DP have been obtained. Then, for each partition p_i, the estimates $\widehat{N}_x^j$ of the number of targets for each cell W_j^i are calculated as

$$\widehat{N}_x^j = \arg\max_n p\left(\left|W_j^i\right| \mid N_x^j = n\right). \tag{17}$$

If $\widehat{N}_x^j$ is larger than one, split the cell W_j^i into $\widehat{N}_x^j$ smaller cells, denoted by $\{W_s^+\}_{s=1}^{\widehat{N}_x^j}$ (Granström et al. [22] use K-means++ clustering to split the measurements in the cell). Then add a new partition, consisting of the new cells along with the other cells in p_i, to the list of partitions obtained by DP. For simplicity, DP-SP is short for the partition method whose partitions are obtained by the distance partitioning with subpartitioning.

4. The Proposed ET-GMPHD Tracker

The trajectories of extended targets were not provided in the original ET-GMPHD filter [21, 22]. This section describes

the proposed ET-GMPHD tracker which can provide the trajectories of individual extended targets according the state estimates of extended targets and their labels. It assigns the labels to the Gaussian terms of the mixture representing the posterior intensity function and evolves these labels through time without affecting the ET-GMPHD tracker recursion. This idea is inspired by the GMPHD tracker proposed in [32] which only adapts to point targets. Here, we extend it to the extended targets and achieve the ET-GMPHD tracker, in which the update step is different from that of the GMPHD tracker. Moreover, the method of the label processing when Gaussian terms of the mixture are merged was not provided. It will be also discussed in this section.

4.1. Label Evolvement for the ET-GMPHD Tracker. At time step $k = 0$, a unique label is assigned to each Gaussian term of the intensity function v_0

$$v_0(\mathbf{x}) = \sum_{i=1}^{J_0} \omega_0^{(i)} \mathcal{N}\left(\mathbf{x}; m_0^{(i)}, P_0^{(i)}\right) \tag{18}$$

to form the set

$$\mathcal{L}_0 = \left\{l_0^{(1)}, \ldots, l_0^{(J_0)}\right\}, \tag{19}$$

where $l_0^{(j)}$ denotes the label of jth Gaussian term with mean $m_0^{(j)}$ and covariance $P_0^{(j)}$.

The structure of propagating the Gaussian term and its label evolvement is shown as in Figure 1.

Given the posterior intensity $v_{k-1}(\mathbf{x})$ at time step $k-1$

$$v_{k-1}(\mathbf{x}) = \sum_{i=1}^{J_{k-1}} \omega_{k-1}^{(i)} \mathcal{N}\left(\mathbf{x}; m_{k-1}^{(i)}, P_{k-1}^{(i)}\right), \tag{20}$$

the predicted intensity at time step k is also a Gaussian mixture and can be expressed as

$$v_{k|k-1}(\mathbf{x}) = v_{S,k|k-1}(\mathbf{x}) + v_{\beta,k|k-1}(\mathbf{x}) + \eta_k(\mathbf{x}), \tag{21}$$

where

$$v_{S,k|k-1}(\mathbf{x}) = p_{S,k} \sum_{i=1}^{J_{k-1}} \omega_{k-1}^{(i)} \mathcal{N}\left(\mathbf{x}; m_{S,k|k-1}^{(i)}, P_{S,k|k-1}^{(i)}\right),$$

$$m_{S,k|k-1}^{(i)} = \mathbf{F}_{k-1} m_{k-1}^{(i)},$$

$$P_{S,k|k-1}^{(i)} = \mathbf{Q}_{k-1} + \mathbf{F}_{k-1} P_{k-1}^{(i)} \left(\mathbf{F}_{k-1}\right)^T,$$

$$v_{\beta,k|k-1}(\mathbf{x}) = \sum_{i=1}^{J_{k-1}} \sum_{j=1}^{J_{\beta,k}} \omega_{k-1}^{(i)} \omega_{\beta,k}^{(j)} \mathcal{N}\left(\mathbf{x}; m_{\beta,k|k-1}^{(i,j)}, P_{\beta,k|k-1}^{(i,j)}\right), \tag{22}$$

$$m_{\beta,k|k-1}^{(i,j)} = \mathbf{F}_{\beta,k-1}^{(j)} m_{k-1}^{(i)},$$

$$P_{\beta,k|k-1}^{(i,j)} = \mathbf{Q}_{\beta,k-1}^{(j)} + \mathbf{F}_{\beta,k-1}^{(j)} P_{k-1}^{(i)} \left(\mathbf{F}_{\beta,k-1}^{(j)}\right)^T,$$

$$\eta_k(\mathbf{x}) = \sum_{i=1}^{J_{\eta,k}} \omega_{\eta,k}^{(i)} \mathcal{N}\left(\mathbf{x}; m_{\eta,k}^{(i)}, P_{\eta,k}^{(i)}\right).$$

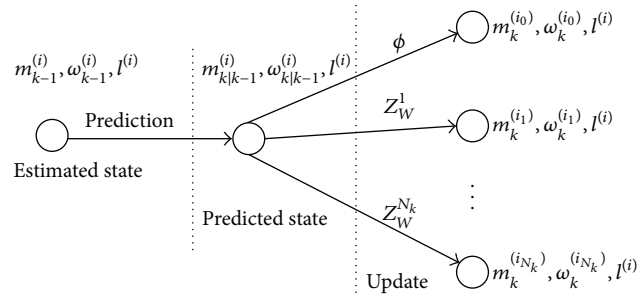


Figure 1: Tree structure of propagating ith Gaussian term and its label evolvement.

We construct the set of labels as follows:

$$\mathcal{L}_{k|k-1} = \mathcal{L}_{k-1} \cup \left\{l_{\eta,k}^{(1)}, \ldots, l_{\eta,k}^{(J_{\eta,k})}\right\} \cup \left\{l_{\beta,k}^{(1,1)}, \ldots, l_{\beta,k}^{(J_{k-1},J_{\beta,k})}\right\}, \tag{23}$$

where $\mathcal{N}(\mathbf{x}; m_{S,k|k-1}^{(i)}, P_{S,k|k-1}^{(i)})$ retains the label of its prior $\mathcal{N}(\mathbf{x}; m_{k-1}^{(i)}, P_{k-1}^{(i)})$, $l_{\eta,k}^{(i)}$ is the new label associated with ith Gaussian term introduced by the birth process, and $l_{\beta,k}^{(i,j)}$ is the new label of jth Gaussian term spawned by ith Gaussian term of the mixture.

The predicted intensity is updated according to (6). Each term in the predicted Gaussian mixture gives rise to $(1 + N_k)$ terms in the updated mixture. N_k is the number of the cells in all the partitions of the measurement set Z_k, as shown as follows:

$$N_k = \sum_{i=1}^{N_p} N_W^{(i)}, \tag{24}$$

where N_P is the number of the partitions of the measurement set Z_k and $N_W^{(i)}$ is the number of the cells in ith partition. We assign the same label to each of the updated Gaussian terms as its associated predicted term. As shown in Figure 1, the Gaussian term $\mathcal{N}(\mathbf{x}; m_k^{(i_j)}, P_k^{(i_j)})$ $(0 \le j \le N_k)$ gets the same label as that of $\mathcal{N}(\mathbf{x}; m_{k|k-1}^{(i)}, P_{k|k-1}^{(i)})$. As a result, we can obtain a number of updated Gaussian terms with different weights for every predicted Gaussian term.

As shown in Figure 1, each tree has its unique label that is the same as the label of the Gaussian term at its root. Each branch of a tree is a possible track of a target. The likelihood of each track is given by its weight. As time goes on, the number of Gaussian components increases sharply. Thus it is necessary to take measures to keep the number of Gaussian components at a computationally tractable level. After discarding those Gaussian components with weights below a preset truncation threshold T, three steps need to be carried out for each tree (Figure 1). First, the branch with the largest weight, B_L, is found. Second, we need to find those branches, represented as B_N, whose Gaussian components are so close to the branch B_L that they could be approximated by a single Gaussian. Finally, we will merge the branch B_L and the branches B_N into one branch and discard other branches of the tree. After the above three steps, only one branch

Given $\{\omega_k^{(i)}, m_k^{(i)}, P_k^{(i)}, l_k^{(i)}\}_{i=1}^{J_k}$, a truncation threshold T, a merging threshold U and a maximum allowable number of Gaussian terms $J_{\max}$.
Set $n = 0$, and $I = \{i = 1, \ldots, J_k \mid \omega_k^{(i)} > T\}$
Repeat:

$j := \arg\max_{i \in I} \omega_k^{(i)}$

$M := \left\{i \in I \mid l_k^{(i)} == l_k^{(j)}\right\}$

$n := n + 1$

$\widetilde{l}_k^{(n)} = l_k^{(j)}$

$N := \left\{i \in M \mid \left(m_k^{(i)} - m_k^{(j)}\right)^T \left(P_k^{(i)}\right)^{-1} \left(m_k^{(i)} - m_k^{(j)}\right) \leq U\right\}$

$\widetilde{\omega}_k^{(n)} = \sum_{i \in N} \omega_k^{(i)}$

$\widetilde{m}_k^{(n)} = \frac{1}{\widetilde{\omega}_k^{(n)}} \sum_{i \in N} \omega_k^{(i)} m_k^{(i)}$

$\widetilde{P}_k^{(n)} = \frac{1}{\widetilde{\omega}_k^{(n)}} \sum_{i \in N} \omega_k^{(i)} \left(P_k^{(i)} + \left(\widetilde{m}_k^{(n)} - \widetilde{m}_k^{(i)}\right)\left(\widetilde{m}_k^{(n)} - \widetilde{m}_k^{(i)}\right)^T\right)$

$I := I \setminus M$

Until $I = \emptyset$

If $n > J_{\max}$, replace $\{\widetilde{\omega}_k^{(i)}, \widetilde{m}_k^{(i)}, \widetilde{P}_k^{(i)}, \widetilde{l}_k^{(i)}\}_{i=1}^{n}$ by those the $J_{\max}$ Gaussian terms with largest weights.

Output $\{\widetilde{\omega}_k^{(i)}, \widetilde{m}_k^{(i)}, \widetilde{P}_k^{(i)}, \widetilde{l}_k^{(i)}\}_{i=1}^{\widetilde{J}_k}$, where $\widetilde{J}_k = \min(J_{\max}, n)$

Algorithm 1: Pruning and merging algorithm for the ET-GMPHD tracker.

of each tree is obtained, which contains the estimated state and corresponding label. The proposed pruning and merging algorithm are summarized in Algorithm 1.

Let $\{\widetilde{\omega}_k^{(i)}, \widetilde{m}_k^{(i)}, \widetilde{P}_k^{(i)}, \widetilde{l}_k^{(i)}\}_{i=1}^{\widetilde{J}_k}$ denote the remaining Gaussian components after pruning and merging, and the intensity function can be expressed as

$$\widetilde{v}_k(\mathbf{x}) = \sum_{i=1}^{\widetilde{J}_k} \widetilde{\omega}_k^{(i)} \mathcal{N}\left(\mathbf{x}; \widetilde{m}_k^{(i)}, \widetilde{P}_k^{(i)}\right). \tag{25}$$

At time step k, state estimates of individual extended targets and their labels are extracted by picking the means of the Gaussian terms whose weights are greater than a chosen threshold, as shown as follows:

$$\begin{aligned} \widehat{X}_k &= \left\{m_k^{(i)} : \omega_k^{(i)} > \omega_{\mathrm{Th}},\ i = 1, \ldots, \widetilde{J}_k\right\}, \\ \widehat{L}_k &= \left\{l_k^{(i)} : \omega_k^{(i)} > \omega_{\mathrm{Th}},\ i = 1, \ldots, \widetilde{J}_k\right\}. \end{aligned} \tag{26}$$

Thus the trajectories of the targets can be determined directly by the evolution of the Gaussian mixture.

4.2. Track Management Scheme for the ET-GMPHD Tracker. For the ET-GMPHD tracker, to form the tracks of the extended targets, a scheme of initiating, confirming, propagating, and terminating tracks is described below.

4.2.1. Track Initiation. At $k = 0$, initialize a tree with $m_0^{(i)}$ as its root and $l_0^{(i)}$ as its label for $i = 1, \ldots, J_0$. At time step $k > 0$, we initialize a tree for every Gaussian term introduced by new birth process. For the tree, $m_{\eta,k}^{(i)}$ and $l_{\eta,k}^{(i)}$ can be regarded as its root and label, respectively.

4.2.2. Track Confirmation. As mentioned in the preceding section, after pruning and merging algorithm each tree has only one branch. We classify a tree as a confirmed tree, if its merged branch weight satisfies $\omega_k^{(i)} > \omega_{\mathrm{Th}}$ in the past three time steps. A confirmed tree provides one confirmed track whose label is the same as that of the tree it belongs to. All the confirmed tracks form a track set $\mathcal{T}_k = \{T_k^{(i)}\}_{i=1}^{N_k^T}$, where N_k^T is the number of tracks in track set $\mathcal{T}_k$. Each track $T_k^{(i)}$ contains the set of estimated target states $\mathbf{X}_k^{(i)}$, corresponding time steps set $t_k^{(i)}$ and its label $l_k^{(i)}$. $\mathbf{X}_k^{(i)}$ contains all the estimated states of the extended target, whose label is equal to $l_k^{(i)}$, from the time step it appears to the time step k.

4.2.3. Track Propagation and Termination. In order to achieve a good performance of the ET-GMPHD tracker in the presence of the detection uncertainty, an undefined tree set $\mathcal{U}_k = \{U_k^{(i)}\}_{i=1}^{N_k^U}$ is constructed, where N_k^U is the number of undefined trees. If a tree has been confirmed before the time step k and the branch weight $\omega_k^{(i)} \leq \omega_{\mathrm{Th}}$ at current time step k, we consider the tree from k as an undefined tree. If the undefined tree lasts three time steps, its corresponding track is terminated. Otherwise, we combine the undefined tree with its corresponding confirmed tree, and then its corresponding track is propagated. The management scheme of the track

Given:
Output of Algorithm 1, $\{\widetilde{\omega}_k^{(i)}, \widetilde{m}_k^{(i)}, \widetilde{P}_k^{(i)}, \widetilde{l}_k^{(i)}\}_{i=1}^{\widetilde{l}_k}$; Extended targets state estimates $\widehat{X}_k = \{m_k^{(i)}\}_{i=1}^{N_k^{\widehat{X}}}$, and their labels $\widehat{L}_k = \{l_k^{(i)}\}_{i=1}^{N_k^{\widehat{X}}}$;
Track set $\mathscr{T}_{k-1}$, Undefined tree set $\mathscr{U}_{k-1}$,
Candidate terminated track set $\mathscr{C} = \{C^{(i)}\}_{i=1}^{N_C}$.
Propagation and Termination:
$j = 0$, $\mathscr{U}_k = \mathscr{U}_{k-1}$
For $i = 1 : N_k^{\widehat{X}}$
  If $l_k^{(i)} \in \mathscr{L}_{k-1}$ $(l_k^{(i)} == l_{k-1}^{(T,n)})$
    $\{\mathscr{L}_{k-1} = \{l_{k-1}^{(T,n)}\}_{n=1}^{N_{k-1}^T}$: set of the labels of track set $\mathscr{T}_{k-1}\}$
    $j = j + 1$,
    $\mathbf{X}_k^{(j)} = \mathbf{X}_{k-1}^{(n)} \cup m_k^{(i)}$, $t_k^{(j)} = t_{k-1}^{(i)} \cup k$,
    $T_k^{(j)} = \{\mathbf{X}_k^{(j)}, t_k^{(j)}, l_k^{(i)}\}$.
  Else If $l_k^{(i)} \in \mathscr{L}_k^U$ $(l_k^{(i)} == l_k^{(U,n)})$
    $\mathscr{L}_k^U = \{l_k^{(U,n)}\}_{n=1}^{N_k^U}$: set of the labels of $\mathscr{U}_k$
    $\mathscr{U}_k = \mathscr{U}_k \setminus U_k^{(n)}$,
    $j = j + 1$,
    $T_k^{(j)} = C^{(n)} \cup U_k^{(n)} \cup \{m_k^{(i)}, k\}$ $(l^{(C,n)} == l_k^{(i)})$
    $\{\mathscr{L}^C = \{l^{(C,n)}\}_{n=1}^{N_C}$: set of the labels of $\mathscr{C}\}$,
    $\mathscr{C} = \mathscr{C} \setminus C^{(n)}$.
  Else
    Track Initiation and Confirmation Steps.
  End If
End For
If $N_k^U \neq 0$
  $A = \emptyset$,
  For $i = 1 : N_k^U$
    If $N_T^{U_k^{(i)}} == 2$ (i.e. $U_k^{(i)}$ lasts three time steps)
    $\{N_T^{U_k^{(i)}}$ is the time steps that undefined tree $U_k^{(i)}$ lasts$\}$
      $A = A \cup U_k^{(i)}$,
      Terminate the track in $\mathscr{C}$ whose label equals to $l_k^{(U,i)}$.
    Else
      $\mathbf{X}_k^{(U,i)} = \mathbf{X}_k^{(U,i)} \cup \widetilde{m}_k^{(n)}$ $\{l_k^{(U,i)} == \widetilde{l}_k^{(n)}\}$,
      $t_k^{(U,i)} = t_k^{(U,i)} \cup k$, $U_k^{(i)} = \{\mathbf{X}_k^{(U,i)}, t_k^{(U,i)}, l_k^{(U,i)}\}$.
    End If
  End For
  $\mathscr{U}_k = \mathscr{U}_k \setminus A$,
  Update the number of undefined trees N_k^U.
End If
$B = \mathscr{L}_{k-1} \setminus \mathscr{L}_k$,
If $B \neq \emptyset$
  $\mathscr{C} = \mathscr{C} \cup \{$tracks in $\mathscr{T}_{k-1}$ whose labels are the same as $B\}$,
  $j = N_k^U$,
  For $i = 1 : N_B$ $\{N_B$ is the cardinality of set $B\}$
    $j = j + 1$,
    $\mathbf{X}_k^{(U,j)} = \widetilde{m}_k^{(n)}$ $\{\widetilde{l}_k^{(n)} == l_B^i\}$,
    $t_k^{(U,j)} = k$,

Algorithm 2: Continued.

$U_k^{(j)} = \{\mathbf{X}_k^{(U,j)}, t_k^{(U,j)}, l_B^i\}$.
End for
End If

Algorithm 2: Scheme of the track propagation and termination.

Require: Measurement set Z_k
Step 1. Obtain the partitions set $Z^p = \{p_1, \ldots, p_{N_p}\}$ using distance partitioning, where N_p is the number of partitions.
Step 2.
for $i = 1, \ldots, N_p$ do
  for $j = 1, \ldots, |p_i|$ do
    $\widehat{N}_x^j = \arg\max_n p\left(\left|W_j^i\right| \mid N_x^j = n\right)$
    if $\widehat{N}_x^j > 1$, then
      $p_i = p_i \setminus W_j^i$
      $\{W_k^+\}_{k=1}^{\widehat{N}_x^j} = \text{split}\left(\widehat{N}_x^j, W_j^i\right)$
      $p_i = p_i \cup \{W_k^+\}_{k=1}^{\widehat{N}_x^j}$
    end if
  end for
end for

Algorithm 3: Mixture partitioning algorithm.

propagation and termination for the ET-GMPHD tracker is summarized in Algorithm 2.

4.3. Mixture Partitioning. It is obvious that the performance of the ET-GMPHD tracker largely depends on the performance of the update step which depends, to a great extent, on the partitioning algorithm of measurements. When targets are spatially close, so are their position measurements. If the region occupied by measurements from an extended target overlaps with that of the other extended target to a certain degree, the cluster algorithm could not reasonably split the cell which contains measurements from the two extended targets. The weight of the partition, in which the cell containing measurements from the spatially close extended targets is split into the right number cells by SP, may be smaller than that of the partition, in which measurements from spatially close extended targets are put in one cell. Therefore, the partition obtained by SP does not take effect in the original ET-GMPHD filter algorithm. In the case discussed above, the original ET-GMPHD filter with DP-SP still has the problem of underestimation of target set cardinality. If two or more extended targets are considered as one, it would be assigned a new label or one of their labels. When they separate from each other again, they would be also labeled as new.

Since DP [21] works well when extended targets are away from each other, in this case we still employ DP. In the situation that two or more extended targets are spatially close, the main error results from the cardinality error of the target set and the error of the target state estimates is relatively small. The mixture partitioning algorithm is proposed to reduce the cardinality error of the target set. It can be described as follows. After partitioning the measurement set by DP, the number of targets for each cell W_j^i is estimated using (17) for each partition p_i, denoted by $\widehat{N}_x^j$. If $\widehat{N}_x^j > 1$, we split the cell W_j^i into $\widehat{N}_x^j$ small cells by Kernel Fuzzy c-means (KFCM) cluster. Kernel functions in KFCM cluster can map the data in the original space to a high-dimensional feature space, in which we can perform clustering more efficiently than c-means cluster and Fuzzy c-means cluster. The Gaussian function is chosen as the kernel function in this papers. The mixture partitioning algorithm is shown in Algorithm 3.

We will describe how to choose the partitioning method as follows. In practical applications, the extended target state always contains position and velocity $\mathbf{x}_k = [x_k, y_k, v_k^x, v_k^y]^T$. The measurement is mainly for position component, sometimes velocity component included. Here two situations are discussed. One is that the measurement only contains the position component, and the other is that the measurement contains not only the position component but also the velocity component.

According to the estimated extended target state set $\widehat{X}_{k-1}$ at time step $k-1$, we calculate the predicted target state set $\widehat{X}_{k|k-1} = \{\mathbf{x}_{k|k-1}^{(i)}\}_{i=1}^{N_{x,k|k-1}}$ by prediction and then obtain the corresponding estimated measurement set $\widehat{\mathbf{Z}}_k = \{\widehat{\mathbf{z}}^{(i)}\}_{i=1}^{N_{x,k|k-1}}$ for the centers of extended targets.

If the measurement only contains the position component, $\widehat{\mathbf{Z}}_k = \{\widehat{\mathbf{z}}^{(i)}\}_{i=1}^{N_{x,k|k-1}}$ is the set of the estimated position measurements of the centers of extended targets. To check

whether there exist extended targets which are spatially close to others, firstly we calculate the Mahalanobis distance between each pair of the estimated position measurements. Then since the smaller the Mahalanobis distance between two estimated position measurements is, the closer the two extended targets are, the following equation

$$\mathbf{d}_M\left(\hat{\mathbf{z}}^{(i)}, \hat{\mathbf{z}}^{(j)}\right) \le \delta_{P_D}, \quad \text{if } \exists i \ne j,\ 1 \le i, j \le N_{x,k|k-1} \tag{27}$$

is applied. If $\hat{\mathbf{Z}}_k$ satisfies (27), there exists spatially close extended targets, and the mixture partitioning is applied to remedy the problem with underestimation of the target number.

As mentioned in Section 3.2, for $\hat{\mathbf{z}}^{(i)}$ and $\hat{\mathbf{z}}^{(j)}$ belonging to the same target, the Mahalanobis distance $\mathbf{d}_M(\cdot,\cdot)$ is χ^2 distributed with degrees of freedom equal to the measurement vector dimension. Using the inverse cumulative χ^2 distribution function, a unitless distance threshold

$$\delta_{P_D} = \text{invchi2}\left(P_D\right) \tag{28}$$

can be computed for a given probability P_D. Simulations illustrate that the good target tracking results are achieved when P_D satisfies the condition $P_D \in [0.8\ \ 0.82]$.

If the measurement contains the velocity component, that is, $\mathbf{Z}_k = \{\mathbf{z}_k^{(i)}\}_{i=1}^{N_{z,k}}$, where $\mathbf{z}_k^{(i)} = [(\mathbf{p}_k^{(i)})^T, (\mathbf{v}_k^{(i)})^T]^T$, to improve the performance of the ET-GMPHD tracker, two parameters ρ_1, ρ_2 are introduced to adjust the weight of the position distance and velocity distance in DP. We adopt the following equation as the distance measure $\mathbf{d}(\cdot,\cdot)$:

$$\mathbf{d}\left(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}\right) = \rho_1^{(i,j)}\mathbf{d}_M\left(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}\right) + \rho_2^{(i,j)}\mathbf{d}_M\left(\mathbf{v}^{(i)}, \mathbf{v}^{(j)}\right), \tag{29}$$

where $\mathbf{d}_M$ is Mahalanobis distance which is unitless and $\rho_1^{(i,j)}$, $\rho_2^{(i,j)}$ are defined as follows:

$$\begin{aligned}\rho_1^{(i,j)} &= \frac{\chi^2\left(\mathbf{d}_M\left(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}\right)\right)}{\chi^2\left(\mathbf{d}_M\left(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}\right)\right) + \chi^2\left(\mathbf{d}_M\left(\mathbf{v}^{(i)}, \mathbf{v}^{(j)}\right)\right)},\\ \rho_2^{(i,j)} &= \frac{\chi^2\left(\mathbf{d}_M\left(\mathbf{v}^{(i)}, \mathbf{v}^{(j)}\right)\right)}{\chi^2\left(\mathbf{d}_M\left(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}\right)\right) + \chi^2\left(\mathbf{d}_M\left(\mathbf{v}^{(i)}, \mathbf{v}^{(j)}\right)\right)}.\end{aligned} \tag{30}$$

In (29), the distance measure $\mathbf{d}(\cdot,\cdot)$ becomes unitless by means of the Mahalanobis distance. $\rho_1^{(i,j)}$, $\rho_2^{(i,j)}$ regulate the ratio of the position distance and velocity distance in the distance measure. When the position distance and velocity distance of two targets are relatively large, their values of the χ^2 distribution function is nearly to 1, and ρ_1 is roughly equal to ρ_2. When the position distance is small and the velocity distance is relatively large, that is, the two targets are spatially close and their velocities are very different, we mainly use the velocity components to partition the measurement set and vice versa. If there exists the extended targets which are spatially close and whose velocity is similar, that is, the following two equations hold

$$\begin{aligned}&\mathbf{d}_M\left(\mathbf{p}^{(i)}, \mathbf{p}^{(j)}\right) \le \delta_{P_{D1}}, \quad \text{if } \exists i \ne j,\ 1 \le i, j \le N_{x,k|k-1},\\ &\mathbf{d}_M\left(\mathbf{v}^{(i)}, \mathbf{v}^{(j)}\right) \le \delta_{P_{D2}}, \quad \text{if } \exists i \ne j,\ 1 \le i, j \le N_{x,k|k-1},\end{aligned} \tag{31}$$

we employ the mixture partitioning algorithm.

5. Simulation Results

Section 5.1 presents the simulation setup. A comparison of the ET-GMPHD filter with DP and DP-SP is presented in Section 5.2, and the label assignment results based on the original ET-GMPHD filter are shown. In Section 5.3 a comparison between the ET-GMPHD filter with DP-SP and that with mixture partitioning algorithm is described. Finally, the simulation results of the proposed ET-GMPHD tracker are illustrated in Section 5.4.

5.1. Simulation Setup

5.1.1. Target Tracking Setup. We consider a two-dimensional scenario over the surveillance region $[-1000, 1000] \times [-1000, 1000]$ (in m). The state $\mathbf{x}_k = [x_k, y_k, v_k^x, v_k^y]^T$ of each extended target consists of its position (x_k, y_k) and velocity (v_k^x, v_k^y). $[\cdot]^T$ denotes a transpose of a matrix $[\cdot]$. The sensor measurements are given in batches of Cartesian x and y coordinates as shown as follows:

$$\mathbf{z}_k^j = \left[x_k^j, y_k^j\right]^T. \tag{32}$$

Reference back to (1) and (2) here. The parameters in the dynamic and measurement models are shown as follows:

$$\begin{aligned}&\mathbf{F}_k = \begin{bmatrix}1 & 0 & T & 0\\ 0 & 1 & 0 & T\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}, \qquad \mathbf{G}_k = \begin{bmatrix}\frac{T^2}{2} & 0\\ 0 & \frac{T^2}{2}\\ T & 0\\ 0 & T\end{bmatrix},\\ &\mathbf{H}_k = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, \qquad \mathbf{Q}_k = \sigma_{\mathbf{w}}^2\mathbf{I}_2, \qquad \mathbf{R}_k = \sigma_{\mathbf{v}}^2\mathbf{I}_2\end{aligned} \tag{33}$$

with sampling time $T = 1$ s. Here, $\sigma_{\mathbf{w}} = 2$ m/s and $\sigma_{\mathbf{v}} = 20$ m.

The probability of survival is set to $p_S = 0.99$, and the probability of detection is $p_D = 0.99$. In each simulation, clutters are generated with a Poisson rate of 10 clutter measurements per scan, and each target generates measurements with a Poisson rate of 10 measurements per scan. The birth intensity in the simulations is

$$\upsilon_b(\mathbf{x}) = 0.1\mathcal{N}\left(\mathbf{x}; m_b^1, P_b\right) + 0.1\mathcal{N}\left(\mathbf{x}; m_b^2, P_b\right), \tag{34}$$

where

$$\begin{aligned}m_b^1 &= [250, 250, 0, 0]^T,\\ m_b^2 &= [-250, -250, 0, 0]^T,\\ P_b &= \text{diag}\left([100, 100, 25, 25]\right).\end{aligned} \tag{35}$$

The spawn intensity is

$$\upsilon_\beta(\mathbf{x}) = 0.05\mathcal{N}\left(\mathbf{x}; \xi, P_\beta\right), \tag{36}$$

where $P_\beta = \text{diag}([100, 100, 400, 400])$ and ξ is the state of the target from which the new target is spawned.

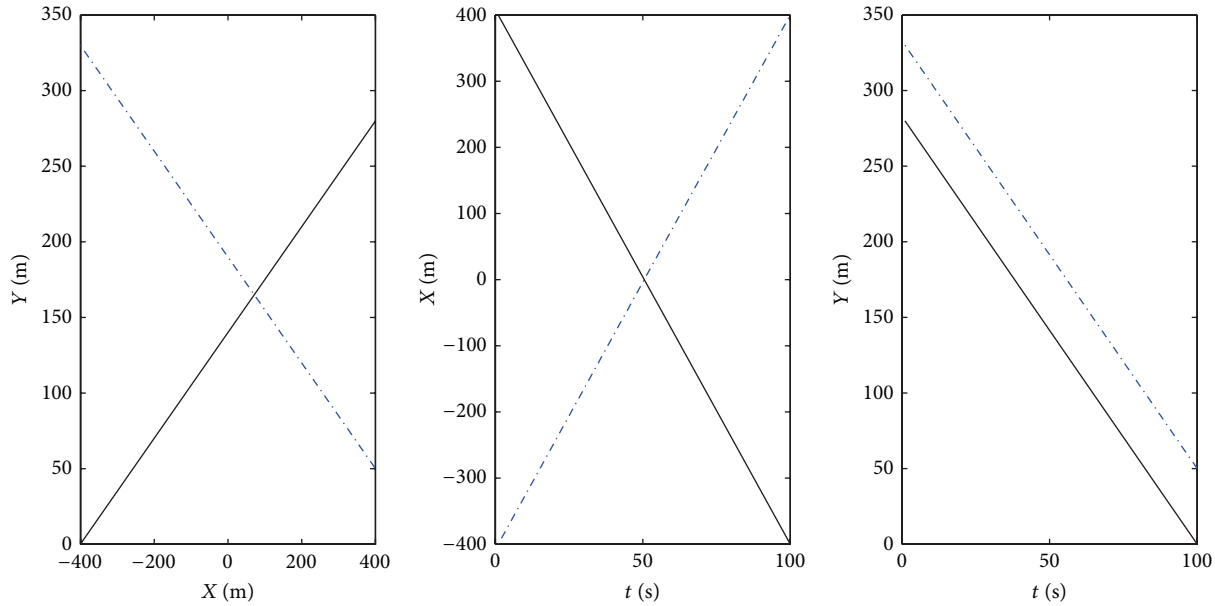


Figure 2: (a) Trajectories of extended targets for the first scenario. (b) The x positions changing with time. (c) The y positions changing with time.

In many practical applications (e.g., radar, laser, and stereo vision), the sensor provides range r and azimuth angle φ given by

$$\mathbf{z}_k^j = \left[r_k^j, \varphi_k^j\right]^T. \tag{37}$$

The work here could be applied to such case by using the appropriate polar to Cartesian conversion equations or by employing the unscented Kalman filter in the update step of the ET-GMPHD tracker.

5.1.2. True Target Track. Three different scenarios are employed in several simulations. The first two of them have two targets. The true x, y positions and their corresponding positions changing with time in the first scenario are shown in Figure 2. The trajectories of two targets in the second scenario are very similar to that in the first scenario: (1) the position of one target is the same with that of the target shown by black solid line in Figure 2; (2) the x position of the other is the same as that of the target shown by blue dotted line in Figure 2(b), and at each time step the y position is 10 m less than that of the target shown by blue dotted line in Figure 2(c). As shown in Figure 2(c), the distance of the two targets in the two scenarios does not change all the time in y direction. The distance of the two targets in y direction is 50 m in the first scenario and 40 m in the second scenario. In the third scenario there are five targets in total. The true x, y positions and their corresponding positions changing with time are shown in Figure 3.

In those simulations, targets generate measurements with standard deviation 20 m in both x and y directions. Thus, a measure of target extent can be considered as the two standard deviation measurement covariance circles with radius 40 m. In all scenarios these circles partly overlap in the case that the targets are close to each other.

5.2. ET-GMPHD Label Assignment. As mentioned in Section 4, the original ET-GMPHD [22] still has the problem that it could not distinguish those extended targets which are spatially close. In this subsection, the first two scenarios described in Section 5.1 are considered to illustrate the problem. The average number of extended targets estimated by the ET-GMPHD algorithm by using DP and DP-SP for two scenarios is given in Figures 4(a) and 4(b), which are obtained by averaging over 100 Monte Carlo runs, respectively.

As shown in Figure 4, though using SP would slightly increase the performance of ET-GMPHD filter, the original ET-GMPHD filter still has the problem of the underestimation of target set cardinality when there are spatially close extended targets. The problem would cause the error label assignment. To illustrate this, target tracks given by the tracker, which employs label assignment and track management scheme discussed in Section 4, based on the original ET-GMPHD filter by using DP-SP for the second scenario are shown in Figure 5. The results indicate that the identities of the extended targets change at time steps 45–55. This is because the ET-GMPHD tracker considers two extended target as one at time steps 47–55. There are four estimated tracks with two true tracks and the labels of estimated tracks are not correct from time step 47 on. For instance, the tracker assigns new labels from time step 56 and takes them as two new extended targets.

5.3. Mixture Partitioning. In this section, we present results that compare the performance of the ET-GMPHD filter

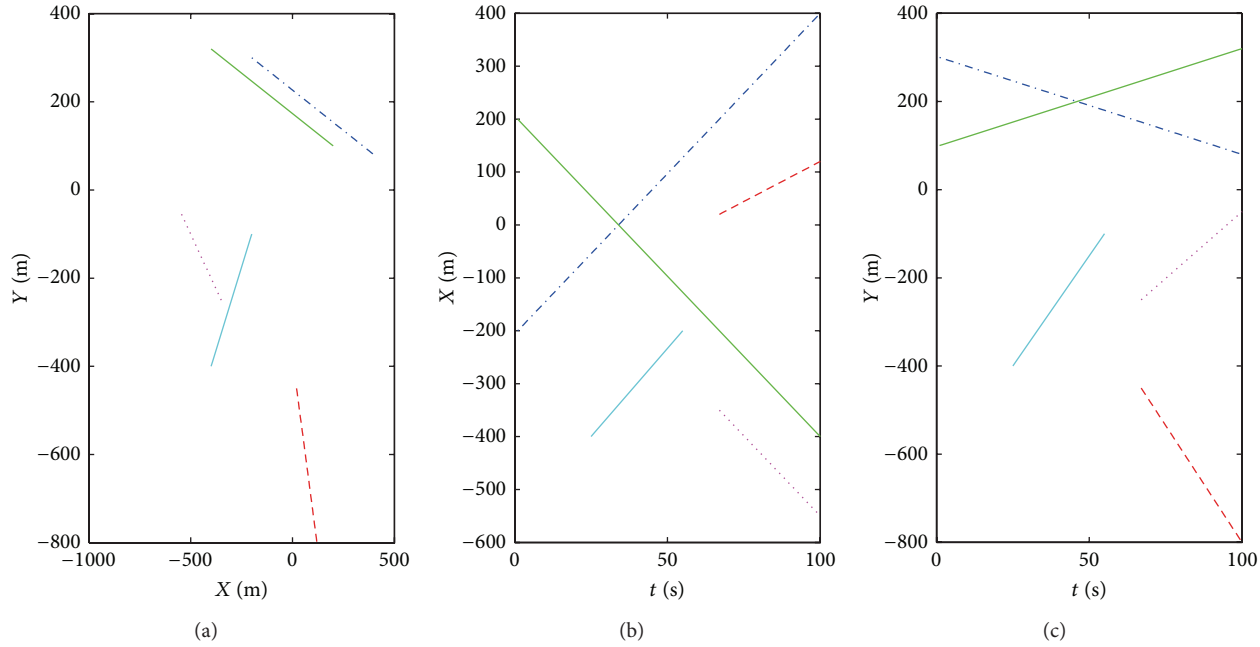


FIGURE 3: (a) Trajectories of extended targets for the third scenario. (b) The x positions changing with time. (c) The y positions changing with time.

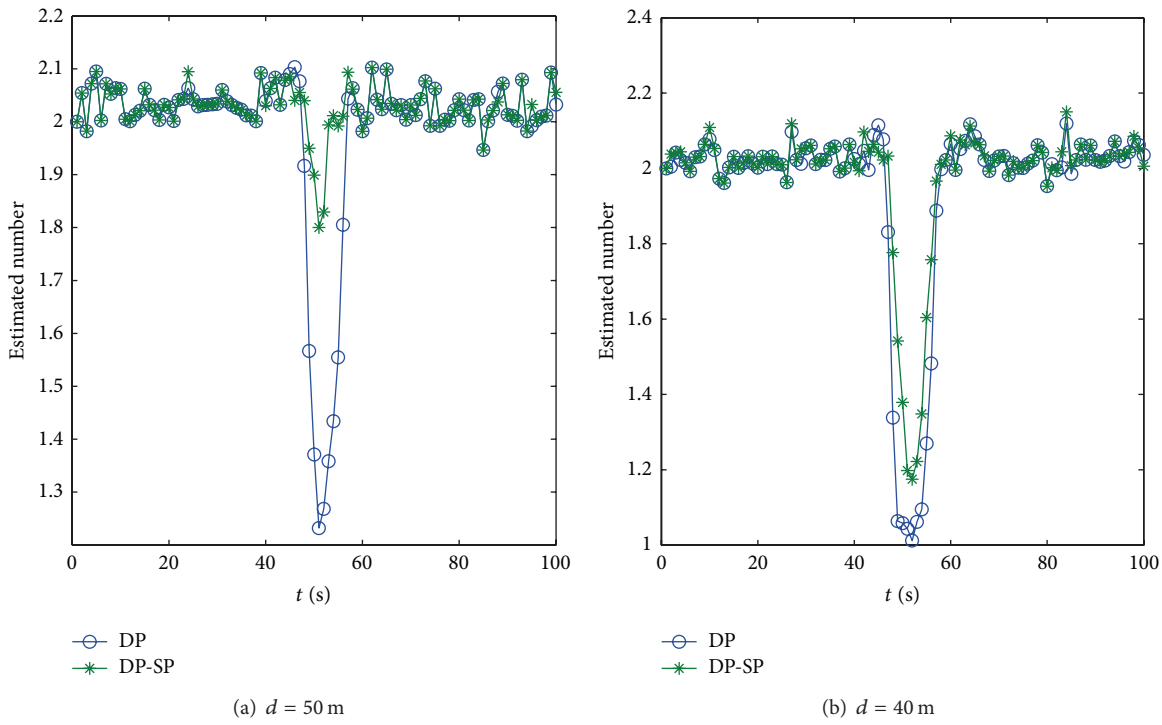


FIGURE 4: Estimated number of extended targets for two scenarios, compared with different partitioning methods (blue circle is only DP; green star shows DP-SP).

by using mixture partitioning and DP-SP. The estimated number of extended targets for the second scenario in Section 5.1 is obtained by averaging over 100 Monte Carlo runs and illustrated in Figure 6. As shown in Figure 6, the average estimated number of extended targets estimated by ET-GMPHD filter by using mixture partitioning is closer to the true number of targets.

5.4. Evaluation of the Proposed ET-GMPHD Tracker. In this section, simulation results are given to show the performance

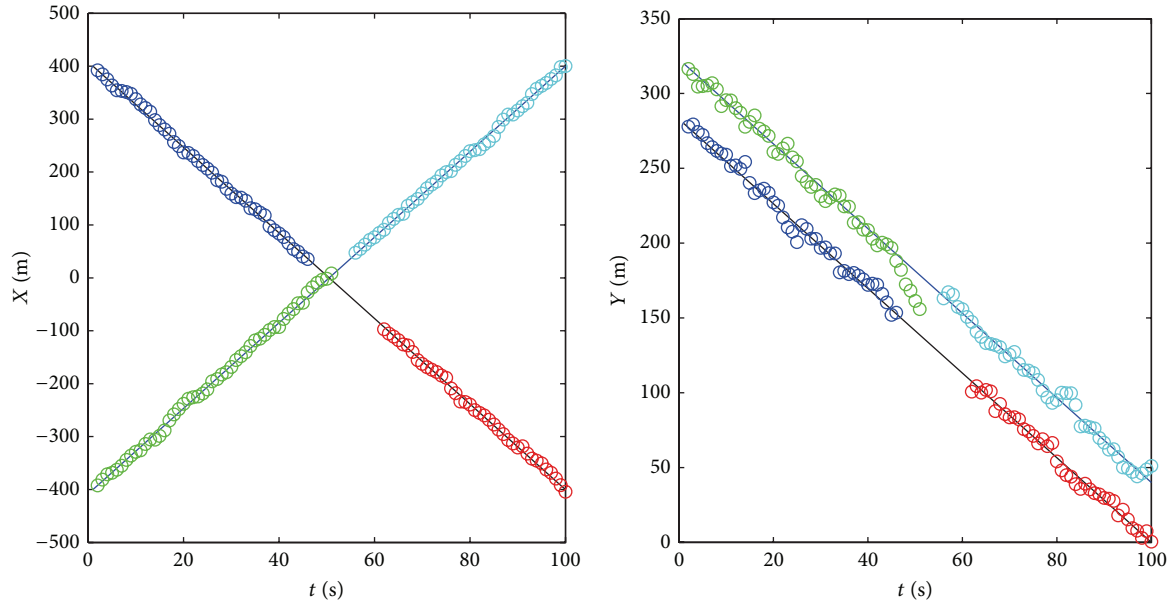


Figure 5: Tracks given by the tracker based on the original ET-GMPHD filter for the second scenario. (a) The x positions changing with time. (b) The y positions changing with time (lines are the true positions and circles are the estimated positions. Different color denotes different track).

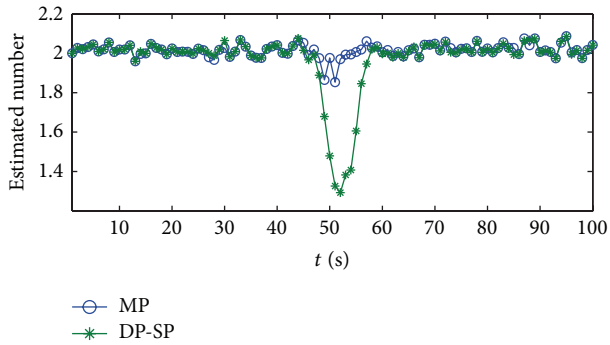


Figure 6: Estimated number of extended targets for the second scenario, compared with different partitioning methods (blue circle is mixture partitioning (MP); green star shows DP-SP).

of the proposed ET-GMPHD tracker and illustrate the effectiveness of the label evolution algorithm and the track management scheme. The last two scenarios in Section 5.1 are considered. The results of the proposed ET-GMPHD tracker for two scenarios are shown in Figures 7 and 8, which show that the proposed ET-GMPHD tracker could give good estimates of the true target trajectories and the estimates are almost free of false tracks.

The results in Figures 7 and 8 are acquired from only one simulation. To evaluate the proposed ET-GMPHD tracker, we choose the optimal subpattern assignment (OSPA) metric as the metric in this paper. Schuhmacher and Xia proposed firstly the OSPA metric in the point process [35] and demonstrated the OSPA metric in performance evaluation of multitarget filtering algorithms [36]. The OSPA metric was used in the multitarget tracking application by Ristic et al. [37]. It is an appropriate metric for multitarget tracking application: it incorporates both the cardinality error and the spatial distance of points and combines various aspects of tracking performance into a single metric. The brief introduction is given in the following.

Two track sets at t_k are represented by

$$\begin{aligned} \mathscr{A}_k &= \{(\ell_1, \mathbf{x}_{k,1}), \ldots, (\ell_m, \mathbf{x}_{k,m})\}, \\ \mathscr{B}_k &= \{(s_1, \mathbf{y}_{k,1}), \ldots, (s_n, \mathbf{y}_{k,n})\}, \end{aligned} \tag{38}$$

where m, n are the cardinalities of sets $\mathscr{A}_k$ and $\mathscr{B}_k$, respectively. For $m \leq n$, the OSPA distance between $\mathscr{A}_k$ and $\mathscr{B}_k$ is defined as

$$\begin{aligned} &D_{p,c}(\mathscr{A}_k, \mathscr{B}_k) \\ &= \left[\frac{1}{n}\left(\min_{\pi \in \Pi_n} \sum_{i=1}^{m} \left(d_c(\widetilde{x}_{k,i}, \widetilde{y}_{k,\pi(i)})\right)^p + (n-m)\cdot c^p\right)\right]^{1/p}, \end{aligned} \tag{39}$$

where $\widetilde{x}_{k,i} = (\ell_i, \mathbf{x}_{k,i})$, $\widetilde{y}_{k,\pi(i)} = (s_\pi(i), \mathbf{y}_{k,\pi(i)})$; $d_c(\widetilde{x}, \widetilde{y}) = \min(c, d(\widetilde{x}, \widetilde{y}))$ is the cutoff distance between two tracks at t_k, with $c > 0$ being the cutoff parameter; $d(\widetilde{x}, \widetilde{y})$ is the base distance between two tracks at t_k; Π_n represents the set of permutations of length m with elements taken from $\{1, 2, \ldots, n\}$; $1 \leq p < \infty$ is the OSPA metric order parameter.

For the case $m > n$, the definition is $D_{p,c}(\mathscr{A}_k, \mathscr{B}_k) = D_{p,c}(\mathscr{B}_k, \mathscr{A}_k)$.

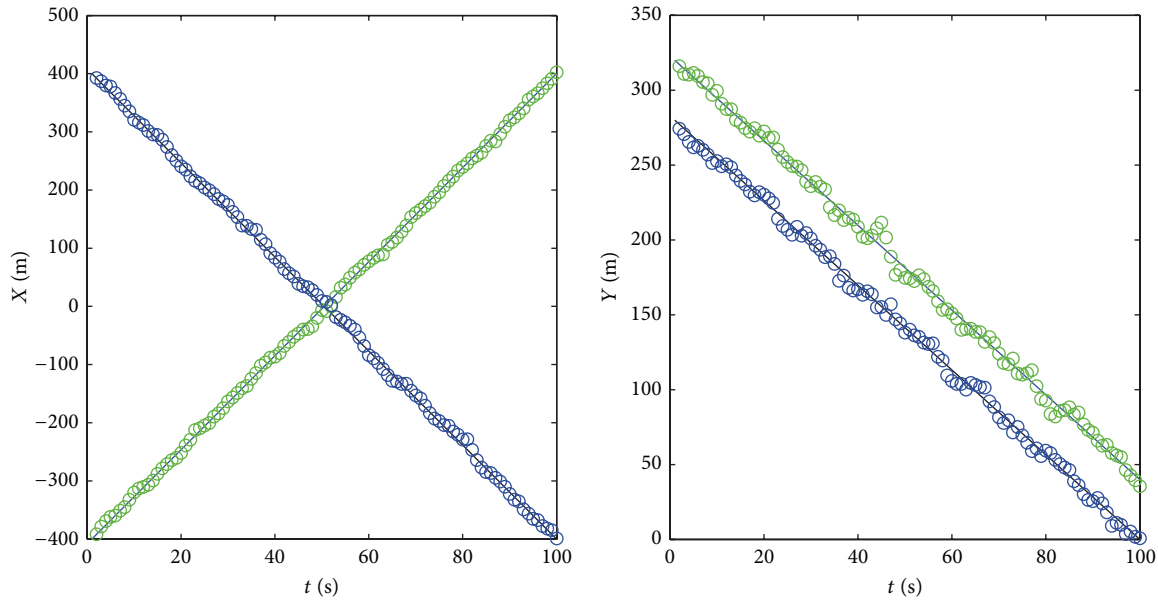


FIGURE 7: Tracks given by the proposed ET-GMPHD tracker for the second scenario. (a) The x positions changing with time. (b) The y positions changing with time (lines are the true positions and circles are the estimated positions. Different color denotes different track).

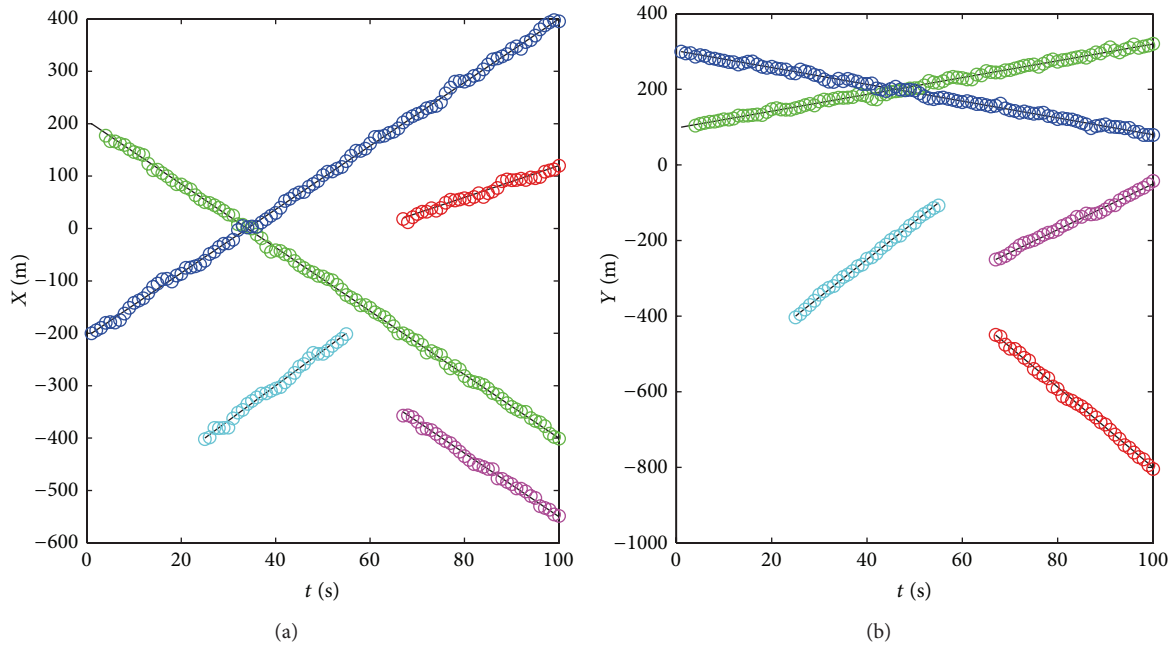


FIGURE 8: Tracks given by the proposed ET-GMPHD tracker for the third scenario. (a) The x positions changing with time. (b) The y positions changing with time (lines are the true positions and circles are the estimated positions. Different color denotes different track).

The base distance $d(\tilde{x}, \tilde{y})$ between tracks $\tilde{x} = (\ell, \mathbf{x})$ and $\tilde{y} = (s, \mathbf{y})$ is defined as

$$d(\tilde{x}, \tilde{y}) = \left(d(\mathbf{x}, \mathbf{y})^{p'} + d(\ell, s)^{p'} \right)^{1/p'}, \tag{40}$$

where $d(\mathbf{x}, \mathbf{y}) = \|x - y\|_{p'}$ is the localization base distance and $d(\ell, s) = \alpha \bar{\delta}(\ell, s)$ is the labeling error distance. $\bar{\delta}(i, j)$ is the complement of the Kronecker delta; that is, $\bar{\delta}(i, j) = 0$, if $i = j$ and $\bar{\delta}(i, j) = 1$, otherwise. Parameter $\alpha \in [0, c]$ controls the penalty assigned to the labeling error $d(\ell, s)$ interpreted relative to the localization distance $d(\mathbf{x}, \mathbf{y})$. The case $\alpha = 0$ assigns no penalty and $\alpha = c$ assigns the maximum penalty.

More details about the OSPA metric can be obtained in [37]. The choice of parameters c and p follows the guidelines

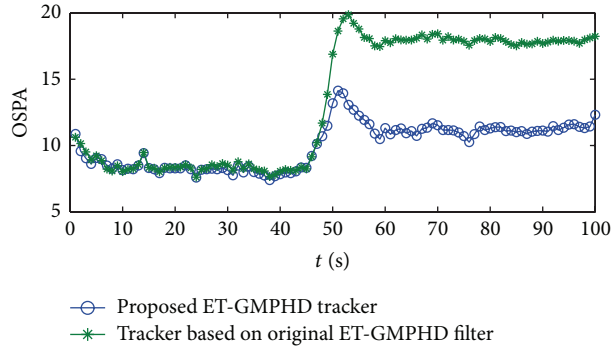
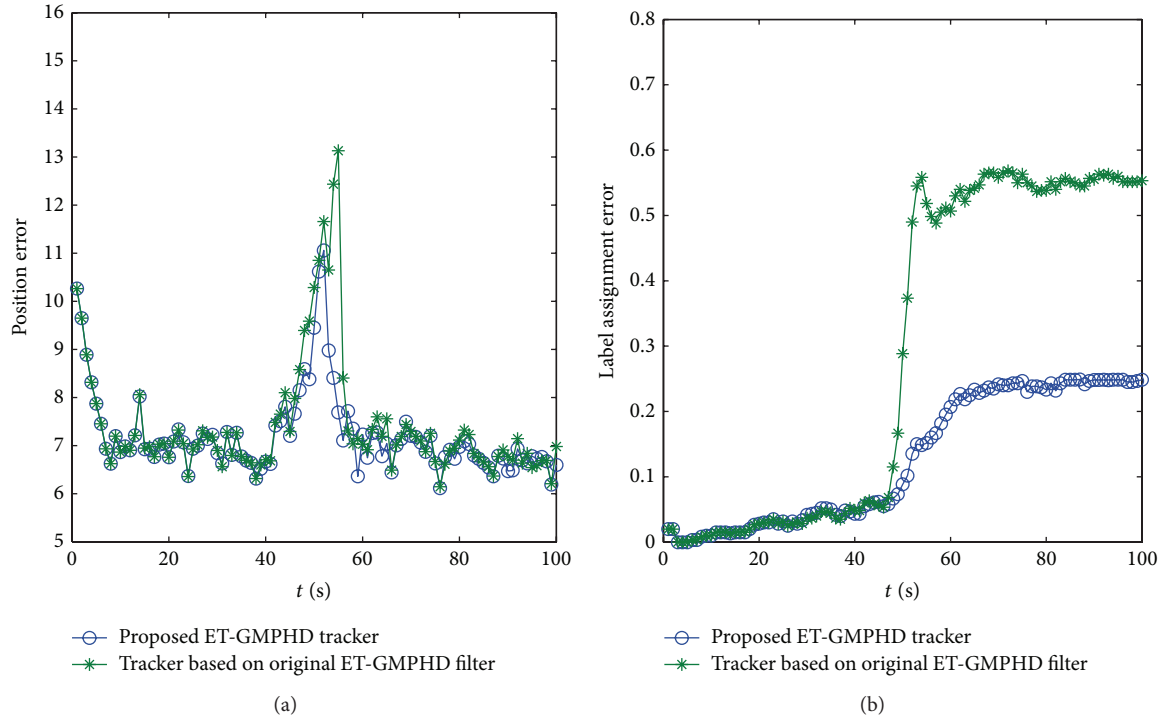


Figure 9: OSPA metric (blue circle is the proposed ET-GMPHD tracker; green star shows the tracker based on the original ET-GMPHD filter by using DP-SP).



(a) (b)

Figure 10: Position and label assignment error (blue circle is the proposed ET-GMPHD tracker; green star shows the tracker based on the original ET-GMPHD filter by using DP-SP).

in [36]. In this paper we set the parameters of the OSPA metric as follows: $c = 25$ m, $p = p' = 1$, and $\alpha = 25$ m. In Figures 9 and 10, the curves are obtained by averaging over 100 Monte Carlo runs.

To illustrate the performance of proposed ET-GMPHD tracker more intuitively, for $m \leq n$ we define the position error and label assignment error as follows:

$$e_p = \left[\frac{1}{m}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m}\left(\min\left(d\left(x_{k,i}, y_{k,\pi(i)}\right), c\right)\right)^{p'}\right)\right]^{1/p'},$$
$$e_l = \frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m}\bar{\delta}\left(\ell_i, s_{\pi(i)}\right)\right), \tag{41}$$

where the distance $d(x, y)$ is the same with that in (40). The case $m > n$ is a trivial modification of (41). The OSPA, position error, and label assignment error curves for the first scenario are shown in Figures 9 and 10.

Figures 9 and 10 show that the proposed ET-GMPHD tracker significantly outperforms the tracker based on the original ET-GMPHD filter, and the proposed label processing and management scheme for tracker are reasonable. The position error of the proposed ET-GMPHD tracker is less than that of the tracker based on the original ET-GMPHD filter by using DP-SP in the situations, where the two extended targets are spatially close to each other. The main difference between two methods is the probability of error label assignment. With the target extension being a circle of 40 m radius and the measurements overlapping significantly at 20 m distance, the

probability of the incorrect label assignment of the proposed method is below 0.25, and the other is beyond 0.6.

6. Conclusion

This paper presents an ET-GMPHD filter-based multiextended-target tracker, which can give the tracks of extended targets according to their labels and the estimation of their states. The proposed ET-GMPHD tracker assigns labels to Gaussian terms and propagates these labels through time without affecting the ET-GMPHD recursion. In addition, a track management scheme for track initiation, track confirmation, track propagation, and track termination is proposed to obtain the tracks of individual extended targets. This paper also presents mixture partitioning algorithm to maintain separate track identities of the extended targets in close proximity and enhance the performance of the ET-GMPHD tracker.

In our future research, we will consider the shape of the extended target and the solution to track the nonlinear models and high maneuvering extended targets.

Conflict of Interests

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

Acknowledgment

This work was supported by National Natural Science Foundation of China (61203220, 61221063, 61074176, and 61370037).

References

[1] K. Gilholm and D. Salmond, "Bayesian approach to extended object and cluster tracking using random matrices spatial distribution model for tracking extended objects," *IEE Proceedings Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, 2005.

[2] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, "Poisson models for extended target and group tracking," in *Signal and Data Processing of Small Targets 2005*, vol. 5913, pp. 230–241, San Diego, Calif, USA, August 2005.

[3] Y. Boers, H. Driessen, J. Torstensson, M. Trieb, R. Karlsson, and F. Gustafsson, "Track-before-detect algorithm for tracking extended targets," *IEE Proceedings: Radar, Sonar and Navigation*, vol. 153, no. 4, pp. 345–351, 2006.

[4] M. Baum and U. D. Hanebeck, "Random hypersurface models for extended object tracking," in *Proceedings of the 9th IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '09)*, pp. 178–183, Ajman, United Arab Emirates, December 2009.

[5] M. Baum, B. Noack, and U. D. Hanebeck, "Extended object and group tracking with Elliptic Random Hypersurface Models," in *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.

[6] M. Baum and U. D. Hanebeck, "Shape tracking of extended objects and group targets with star-convex RHMs," in *Proceedings of the International Conference on Information Fusion*, pp. 338–345, Chicago, Ill, USA, 2011.

[7] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, 2008.

[8] W. Koch and M. Feldmann, "Cluster tracking under kinematical constraints using random matrices," *Robotics and Autonomous Systems*, vol. 57, no. 3, pp. 296–309, 2009.

[9] M. Feldmann, D. Fränken, and J. W. Koch, "Tracking of extended objects and group targets using random matrices," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1409–1420, 2011.

[10] M. Baum, M. Feldmann, D. Franken, U. D. Hanebeck, and J. W. Koch, "Extended object and group tracking: a comparison of random matrices and random hypersurface models," in *Proceedings of the IEEE ISIF Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF '10)*, Leipzig, Germany, 2010.

[11] K. Granstrom, C. Lundquist, and U. Orguner, "Tracking rectangular and elliptical extended targets using laser measurements," in *Proceedings of the 14th International Conference on Information Fusion (FUSION '11)*, pp. 592–599, IEEE, Chicago, Ill, USA, July 2011.

[12] C. Lundquist, K. Granström, and U. Orguner, "Estimating the shape of targets with a PHD filter," in *Proceedings of the 14th International Conference on Information Fusion (Fusion '11)*, pp. 49–56, Chicago, Ill, USA, July 2011.

[13] H. Y. Zhu, C. Z. Han, and C. Li, "An extended target tracking method with random finite set observations," in *Proceedings of the 14th International Conference on Information Fusion*, pp. 73–78, Chicago, Ill, USA, 2011.

[14] B. Jiu, H. Liu, D. Feng, and Z. Liu, "Minimax robust transmission waveform and receiving filter design for extended target detection with imprecise prior knowledge," *Signal Processing*, vol. 92, no. 1, pp. 210–218, 2012.

[15] M. Wieneke and W. Koch, "Probabilistic tracking of multiple extended targets using random matrices," in *Signal and Data Processing of Small Targets*, vol. 7698 of *Proceedings of SPIE*, Orlando, Fla, USA, April 2010.

[16] M. Baum, B. Noack, F. Beutler, D. Itte, and U. D. Hanebeck, "Optimal Gaussian filtering for polynomial systems applied to association-free multi-target tracking," in *Proceedings of the 14th International Conference on Information Fusion (Fusion '11)*, pp. 1–8, Chicago, Ill, USA, July 2011.

[17] R. Mahler, "PHD filters for nonstandard targets, I: extended targets," in *Proceedings of the 12th International Conference on Information Fusion (FUSION '09)*, pp. 915–921, Seattle, Wash, USA, July 2009.

[18] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*, Artech House, Norwood, Mass, USA, 2007.

[19] U. Orguner, C. Lundquist, and K. Granström, "Extended target tracking with a cardinalized probability hypothesis density filter," in *Proceedings of the 14th International Conference on Information Fusion (Fusion '11)*, July 2011.

[20] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[21] K. Granström, C. Lundquist, and U. Orguner, "A Gaussian mixture PHD filter for extended target tracking," in *Proceedings of the 13th Conference on Information Fusion (Fusion '10)*, pp. 1–8, Edinburgh, UK, July 2010.

[22] K. Granström, C. Lundquist, and O. Orguner, "Extended target tracking using a gaussian-mixture PHD filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 3268–3286, 2012.

[23] K. Granström and U. Orguner, "A PHD filter for tracking multiple extended targets using random matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5657–5671, 2012.

[24] K. Panta, B.-N. Vo, S. Singh, and A. Doucet, "Probability hypothesis density filter versus multiple hypothesis tracking," in *Signal Processing, Sensor Fusion, and Target Recognition XIII*, vol. 5429 of *Proceedings of SPIE*, pp. 284–295, Orlando, Fla, USA, April 2004.

[25] L. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Data association combined with the probability hypothesis density filter for multitarget tracking," in *Signal and Data Processing of Small Targets*, vol. 5428 of *Proceedings of SPIE*, pp. 464–475, Orlando, Fla, USA, April 2004.

[26] E. Dror-Rein and H. B. Mitchell, "A 'soft' assignment algorithm for multiple target tracking," *Signal Processing*, vol. 81, no. 8, pp. 1757–1764, 2001.

[27] D. E. Clark and J. Bell, "Data association for the PHD filter," in *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Melbourne, Australia, 2005.

[28] K. Panta, B. N. Vo, and S. Singh, "Improved probability hypothesis density (PHD) filter for multitarget tracking," in *Proceedings of the 2nd International Conference on Intelligent Sensing and Information Processing (ICISIP '05)*, pp. 213–218, Bangalore, India, January 2005.

[29] D. E. Clark, K. Panta, and B.-N. Vo, "The GM-PHD filter multiple target tracker," in *Proceedings of the 9th International Conference on Information Fusion (FUSION '06)*, Florence, Italy, July 2006.

[30] D. E. Clark, B. N. Vo, and J. Bell, "The gm-phd filter multitarget tracking in sonar images," in *Signal Processing, Sensor Fusion and Target Recognition*, Proceedings of SPIE, Kissimmee, Fla, USA, 2006.

[31] N. T. Pham, W. Huang, and S. H. Ong, "Maintaining track continuity in GMPHD filter," in *Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS '07)*, pp. 1–5, Singapore, December 2007.

[32] K. Panta, D. E. Clark, and B.-N. Vo, "Data association and track management for the gaussian mixture probability hypothesis density filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 3, pp. 1003–1016, 2009.

[33] Y. Zhang and H. Ji, "A novel fast partitioning algorithm for extended target tracking using a Gaussian mixture PHD filter," *Signal Processing*, vol. 93, no. 11, pp. 2975–2985, 2013.

[34] A. Scheel, K. Granström, D. Meissner, S. Reuter, and K. Dietmayer, "Tracking and data segmentation using a GGIW filter with mixture clustering," in *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, 2014.

[35] D. Schuhmacher and A. Xia, "A new metric between point process distributions," *Advances in Applied Probability*, vol. 40, no. 3, pp. 651–672, 2008.

[36] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[37] B. Ristic, B.-N. Vo, D. Clark, and B.-T. Vo, "A metric for performance evaluation of multi-target tracking algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3452–3457, 2011.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 736246, 8 pages
http://dx.doi.org/10.1155/2015/736246

Research Article

Robust Control of the Knee Joint Angle of Paraplegic Patients considering Norm-Bounded Uncertainties

Nilson Moutinho dos Santos,[1] Ruberlei Gaino,[2] Márcio Roberto Covacic,[2] Marcelo Carvalho Minhoto Teixeira,[3] Aparecido Augusto de Carvalho,[3] Edvaldo Assunção,[3] Rodrigo Cardim,[3] and Marcelo Augusto Assunção Sanches[3]

[1]*Engenharia Elétrica, Área de Exatas, Faculdade Ingá, Rodovia PR 317, No. 6114, 87035-510 Maringá, PR, Brazil*
[2]*Department of Electrical Engineering, Urban Planning and Technology Center, State University of Londrina, Rodovia Celso Garcia Cid PR 445, Km 380, 86057-970 Londrina, PR, Brazil*
[3]*Department of Electrical Engineering, Universidade Estadual Paulista (UNESP), Campus de Ilha Solteira, 15385-000 Ilha Solteira, SP, Brazil*

Correspondence should be addressed to Ruberlei Gaino; rgaino@uel.br

Received 4 September 2014; Revised 17 December 2014; Accepted 18 December 2014

Academic Editor: John Gunnar Carlsson

A proposal for the knee position control design of paraplegic patients with functional electrical stimulation (FES) using control systems and considering norm-bounded uncertainties is presented. A state-space representation of the knee joint model of the paraplegic patient with its nonlinearity is also demonstrated. The use of linear matrix inequalities (LMIs) in control systems with norm-bounded uncertainties for asymptotic stability is analyzed. The model was simulated in the Matlab environment. The matrix K of state space feedback was obtained through LMIs.

1. Introduction

The application of electrical stimulation in a person's muscle, more particularly in his motor neurons, causes involuntary contraction of this muscle [1].

In order to obtain a muscle contraction it is necessary that the amplitude (or intensity) and duration of the electrical stimulus are inside specific bounds. Then an action potential (AP) is generated and propagates in both directions of the nerve fiber. Complex mechanisms of electrochemical stimuli occur in the neuromuscular structure causing the process of excitation-contraction coupling responsible for the movement of the leg [2].

The modulation of the force by a number of muscle fibers recruited and the speed of fiber recruitment depends on several parameters. Some of these parameters include the proximity of the nerve fiber and the electrode, the electrode diameter, and the variation of the number of active states of the fibers due the variation of the amplitude or pulse duration. As can be seen in Figure 1, the degree of muscle activation, α, is a nonlinear function. It depends on the duration of the stimulus, d [3].

Muscle is a highly complex nonlinear system, capable of producing the same output for a variety of inputs. A property exploited by physiologically activated muscle is its effort to minimize fatigue [4].

Several researchers have used FES to restore some motion activities of persons with injured spinal cord [5]. However, FES is not yet a regular clinical method, because the amount of effort involved in using actual stimulation systems still outweighs the functional benefits they provide.

One serious problem when using FES is that artificially activated muscles fatigue at a faster rate than those activated by the natural physiological processes. Due to these problems, a considerable effort has been directed toward developing FES systems based on closed loop control.

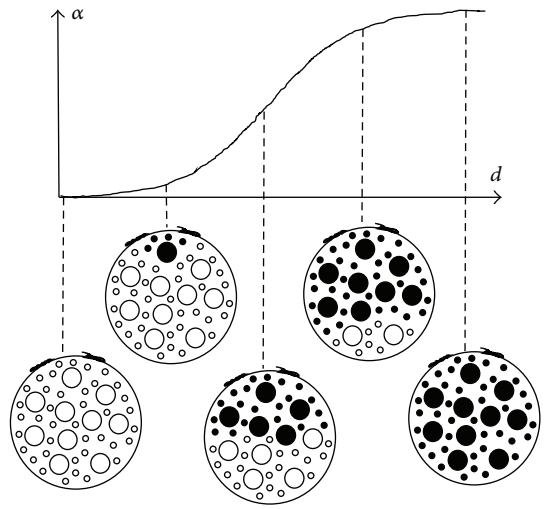


Figure 1: Fibers recruitment curve (the black circles are activated fibers) [3].

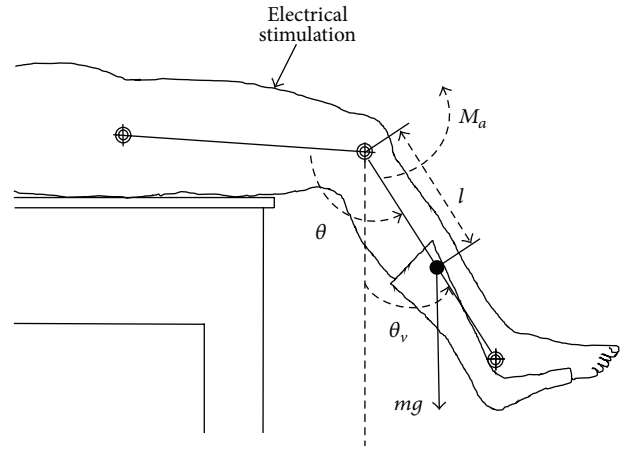


Figure 2: Schematic representation of the patient knee joint plant [6].

Ferrarin and Pedotti developed a dynamic model for the relationship between electrical stimulus and joint torque [5]. In their paper, the dynamics of the lower limb were represented by a nonlinear second-order model, which took into account the gravitational and inertial characteristics of the anatomical segment as well as the damping and stiffness properties of the knee joint.

Considering that when the quadriceps is electrically stimulated its response is nonlinear, Teixeira et al. [7] proposed a nonlinear controller with the aim of controlling the position of the leg of a paraplegic patient [7]. The authors designed a controller to vary the knee joint angle using Takagy-Sugeno fuzzy models. When electrical stimulation is applied to the quadriceps of a patient, using the controller, the knee joint angle can be varied from 0° to 30°. The authors considered the leg mathematical model proposed by [5], and they showed that, for the conditions considered in their experiments, a simple one-pole transfer function was able to model the relationship between stimulus pulse width and active muscle torque [8].

This paper presents a proposal for the leg position control design of paraplegic patients with FES using control systems with uncertainties bounded in norm and a feedback signal obtained from an electrogoniometer which is the most commonly used sensor for measuring the knee joint angle.

As mentioned in [9] "Complex control systems have been recently employed to control the communications among computers, controllers, and sensors due to the enlarging scale of control systems in nowadays applications." Nowadays, it is a very important issue for dynamic systems design.

2. Nonlinear Knee Joint Model

As observed by Gaino et al. in [6], the FES systems based on open-loop control have some shortcomings related to the lack of sensitivity to both external disturbances and changes in internal parameters. On the other hand, FES systems based on closed loop control allow real-time measurement of the motion produced by means of sensors and provide a stimulation pattern with the appropriate modulation [5]. This allows the optimization of the neurostimulator pulse width with consequent reduction of muscle fatigue. Furthermore, Gaino et al. demonstrated in [6] that the feedback signal related to the angle of the knee joint may be obtained by using electrogoniometers.

The chosen plant is the knee joint, and the open kinematic chain is made of two rigid segments: the thigh and shank-foot complex [5], illustrated in Figure 2. It allows us to study the relationship between the major parameters of input stimulation and the torque of the knee joint. In other words, the pulse width with the output represented by the torque of the active knee joint produced by muscles is stimulated under nonisometric conditions [10]. The ankle was fixed using a plastic ankle foot orthosis (AFO), reducing the number of plant's degrees of freedom, causing the length of biarticular muscles to depend only on the position of the knee joint. The thigh was fixed to the support table so that only the dynamics of shank-foot and flexion-extension knee movements were considered. This model was originally proposed in [5].

The positioning of the knee at a desired angle θ between shank and thigh in the sagittal plane is obtained by FES stimulation of the quadriceps muscles [5, 6]. The angle between the shank and the vertical axis in the sagittal plane is θ_v and the active torque produced by FES in the quadriceps is M_a [12].

As in [2, 12], the system's point of interest is $\theta_{v0} = 30^\circ$, and the operating point of the system is not the origin. Thus, according to the stability theory of Lyapunov, it is necessary to make a change of variables to transfer the new operation point to the origin. Thus, the state variables of (1) are given by the following:

(i) $x_1(t) = \Delta\theta_v = \theta_v - \theta_{v0}$, where θ_v is the angle between the shank and the vertical axis in the sagittal plane;

(ii) $x_2 = \dot{x}_1$, the knee joint angular velocity;

Table 1: Parameter values obtained experimentally in [5, 6, 11] to the shank-foot complex of a paraplegic, 30-year-old patient.

	Parameter	Value	Unit
J	Inertial moment of shank-foot complex	0.362	Kg·m^2
m	Mass of shank-foot complex	4.37	Kg
l	Distance between knee and the center of the mass of shank-foot complex	23.8	cm
B	Viscous coefficient	0.27	N·m·s/rad
λ	Coefficient of exponential term	41.208	N·m/rad
E	Coefficient of exponential term	2.024	rad^{-1}
ω	Resting elastic knee angle	2.918	rad
τ	Transfer function coefficient (time constant of the pole)	0.951	s
G	Transfer function coefficient (static gain)	42500	N·m/s

(iii) $x_3 = \Delta M_a = M_a - M_{a0}$, where M_{a0} is the active torque (on quadriceps muscle of the paraplegic patient) in the operation point.

The plant parameters with their meanings and values were obtained experimentally for the case of one patient in [5, 6] and they are shown in Table 1.

The movement of the knee joint resulting from the electrical stimulus applied to the quadriceps is represented by the nonlinear state equation (1), according to Gaino et al. in [2, 12]:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \tilde{f}_{21}(x_1) & -\dfrac{B}{J} & \dfrac{1}{J} \\ 0 & 0 & -\dfrac{1}{\tau} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dfrac{G}{\tau} \end{bmatrix} P_N. \quad (1)$$

The nonlinearity of the mathematical model of the plant comes through the element $a_{21} \rightarrow \tilde{f}_{21}(x_1(t))$ of the characteristic matrix of the system [2, 6, 11, 12], which is given by

$$\tilde{f}_{21}(x_1(t)) = \frac{1}{Jx_1} mgl \sin(x_1(t) + \theta_{v0}) - \frac{1}{Jx_1} \times \left[\lambda \exp\left(-E\left(x_1(t) + \theta_{v0} + \frac{\pi}{2} \right) \right) \cdot \left(x_1(t) + \theta_{v0} + \frac{\pi}{2} - w \right) + M_{a0} \right]. \quad (2)$$

As defined in [2, 6, 11, 12], at the operating point of interest, the following values for the variables are found: $\theta_v = \theta_{v0} = 30^\circ$, and the derivatives $\dot{\theta}_v$ and $\ddot{\theta}_v$ are null. In this case, the active torque of the knee, M_{a0}, produced by electrical stimulation, is given by

$$M_{a0} = mgl \sin(\theta_{v0}) + \lambda \exp\left(-E\left(\theta_{v0} + \frac{\pi}{2} \right) \right) \left(\theta_{v0} + \frac{\pi}{2} - w \right). \quad (3)$$

Considering the values of the parameters of Table 1, taking $g = 9.8$ (m/s^2) and $\theta_{v0} = \pi/6$, and substituting in (3) the value of $M_{a0} = 4.6068$ (N · m) is obtained. In the case where $\theta_{v0} = \pi/3$ (rad), $M_{a0} = 8.7653$ (N · m).

The new input of the system, P_N, is defined from the system input, P, and is known as the pulse width unreferenced [2, 6, 11, 12]. It is given by

$$P_N = P - \frac{M_{a0}}{G}. \quad (4)$$

Since there will only be movement of the paraplegic's leg if an electrical stimulation pulse is applied to the skin of the thigh, that is, $P > 0$, this implies that

$$P_N > -\frac{M_{a0}}{G}. \quad (5)$$

3. Robust Control Systems with Norm-Bounded Uncertainties

In this section the concept of control systems with norm-bounded uncertainties is presented and its methodology is used to design a control angle of the knee joint angle in paraplegic patients discussed in Section 2.

3.1. Systems Control with Norm-Bounded Uncertainties. Let the plant be represented by

$$\dot{x} = (A + \delta A)x + (B + \delta B)u, \quad (6)$$

where the uncertainty matrices are given by

$$\delta A = L\Delta R_A, \quad \delta B = L\Delta R_B, \quad \Delta\Delta^T \preceq I. \quad (7)$$

The matrix Δ is the diagonal matrix of the uncertainties, containing the uncertain parameters normalized on the main diagonal, while the modulus of each is less than or equal to 1. The matrices R_A and R_B are algebraically determined, so that it suits expressions in (6) for δA and δB given in (7).

Theorem 1, that follows the ideas of [13], establishes a sufficient condition for the stability of system (6) in open loop, with the uncertain matrices given in (7) and $u = 0$, in other words, null input.

Theorem 1 (see [13]). *The plant (6) is stable for any matrices δA and δB defined in (7) if there exists a matrix $X = X^T$ and a real constant ϵ that satisfy the following conditions:*

$$\begin{bmatrix} AX + XA^T + \epsilon LL^T & XR_A^T \\ R_A X & -\epsilon I \end{bmatrix} \prec 0, \qquad X \succ 0. \tag{8}$$

Consider, now, the plant (6), with the matrices δA and δB given in (7) and the control law with states feedback:

$$u = -Kx. \tag{9}$$

Thus, considering (6), (7), and the control law (9), the feedback system is defined by

$$\dot{x} = \left[(A - BK) + L\Delta\left(R_A - R_B K\right)\right] x. \tag{10}$$

Theorem 2, that follows the ideas of [13], below corresponds to Theorem 1 but applies to the closed loop system and establishes a sufficient condition for the stability of the feedback system (10), with uncertain matrices given in (7) and input $u = -Kx$.

Theorem 2 (see [13]). *System (10) is stable for any matrices δA and δB defined in (7) if there are matrices $X = X^T$ and W and a real constant ϵ which satisfies the following conditions:*

$$\begin{bmatrix} AX + XA^T - BW - W^T B^T + \epsilon LL^T & XR_A^T - W^T R_B^T \\ R_A X - R_B W & -\epsilon I \end{bmatrix} \prec 0, \tag{11}$$

$$X \succ 0. \tag{12}$$

Based on the solution, the matrix K is given by $K = WX^{-1}$.

3.2. Systems Control with Output Restriction. According to Boyd in [14, 15], stability analysis and control design problems are likely to be described by LMIs, allowing the introduction of other performance indices in the controller design, for example, the specification of the transient response by the decay rate and constraint specifications on the amplitude of control signals and outputs.

A system that offers robust performance should be able to maintain its performance even in the presence of perturbations and disturbances, which are called uncertainties.

Considering a known initial condition $x(0)$ and the LMIs (14), the output $y(t) = Cx(t)$ can be subjected to the constraint (13), for all time $t \geq 0$:

$$\max \left\| y(t) \right\|_2 = \max \sqrt{y^T(t)\, y(t)} \leq \lambda_0. \tag{13}$$

Then the asymptotic stability of system (6), with the output constraint (13), can be carried out by adding the LMIs (14) to (12), keeping $X = X^T \succ 0$, as shown in [16, 17]. Consider

$$\begin{bmatrix} 1 & x(0)^T \\ x(0) & X \end{bmatrix} \geq 0, \qquad \begin{bmatrix} X & XC^T \\ CX & \lambda_0^2 I \end{bmatrix} \geq 0. \tag{14}$$

3.3. Leg Position Control of the Paraplegic Patient. The control of the knee paraplegic patient's position through functional electrical stimulation applied to the quadriceps muscle (see Section 2) is described in [5, 7]. The state space equation is given by (6) with

$$A + \delta A = \begin{bmatrix} 0 & 1 & 0 \\ \widetilde{f}_{21}(x_1) & -\dfrac{B}{J} & \dfrac{1}{J} \\ 0 & 0 & -\dfrac{1}{\tau} \end{bmatrix}, \qquad B + \delta B = \begin{bmatrix} 0 & 0 & \dfrac{G}{\tau} \end{bmatrix}^T. \tag{15}$$

As described in Section 2, the function $\widetilde{f}_{21}(x_1(t))$ is a nonlinearity of the system (see (2)) such that the range of parametric uncertainties is

$$\widetilde{f}_{21\min} \leq \widetilde{f}_{21}\left(x_1(t)\right) \leq \widetilde{f}_{21\max}. \tag{16}$$

In (2) the x_1 variable is at the denominator, and the analysis is more difficult when x_1 is equal to zero. In order to solve this inconvenience, Gaino, in [2], analytically expanded (2) in Taylor series of fifth order. Studying the expanded equation, in a closed interval of interest, Gaino, in [2], found that the approximation error becomes practically zero. The values found by Gaino, considering the operation point $\theta_{v0} = 30^\circ$ and given the range of $-\pi/6 < \theta_v < \pi/6$, are $\widetilde{f}_{21\max} = -21.7834$ and $\widetilde{f}_{21\min} = -36.0086$.

The nominal plant is represented by matrices A and B, given by

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \widetilde{f}_0 & -\dfrac{B}{J} & \dfrac{1}{J} \\ 0 & 0 & -\dfrac{1}{\tau} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 & \dfrac{G}{\tau} \end{bmatrix}^T, \tag{17}$$

where

$$\widetilde{f}_0 = \frac{\widetilde{f}_{21\min} + \widetilde{f}_{21\max}}{2}. \tag{18}$$

The matrices δA and δB, in (6), are decomposed according to (7):

$$L = \begin{bmatrix} 0 & \delta f_{\max} & 0 \end{bmatrix}^T, \qquad R_A = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \qquad \Delta = [\delta], \qquad R_B = [0], \tag{19}$$

where $-1 \leq \delta \leq 1$ and

$$\delta f_{\max} = \frac{\widetilde{f}_{21\max} - \widetilde{f}_{21\min}}{2}. \tag{20}$$

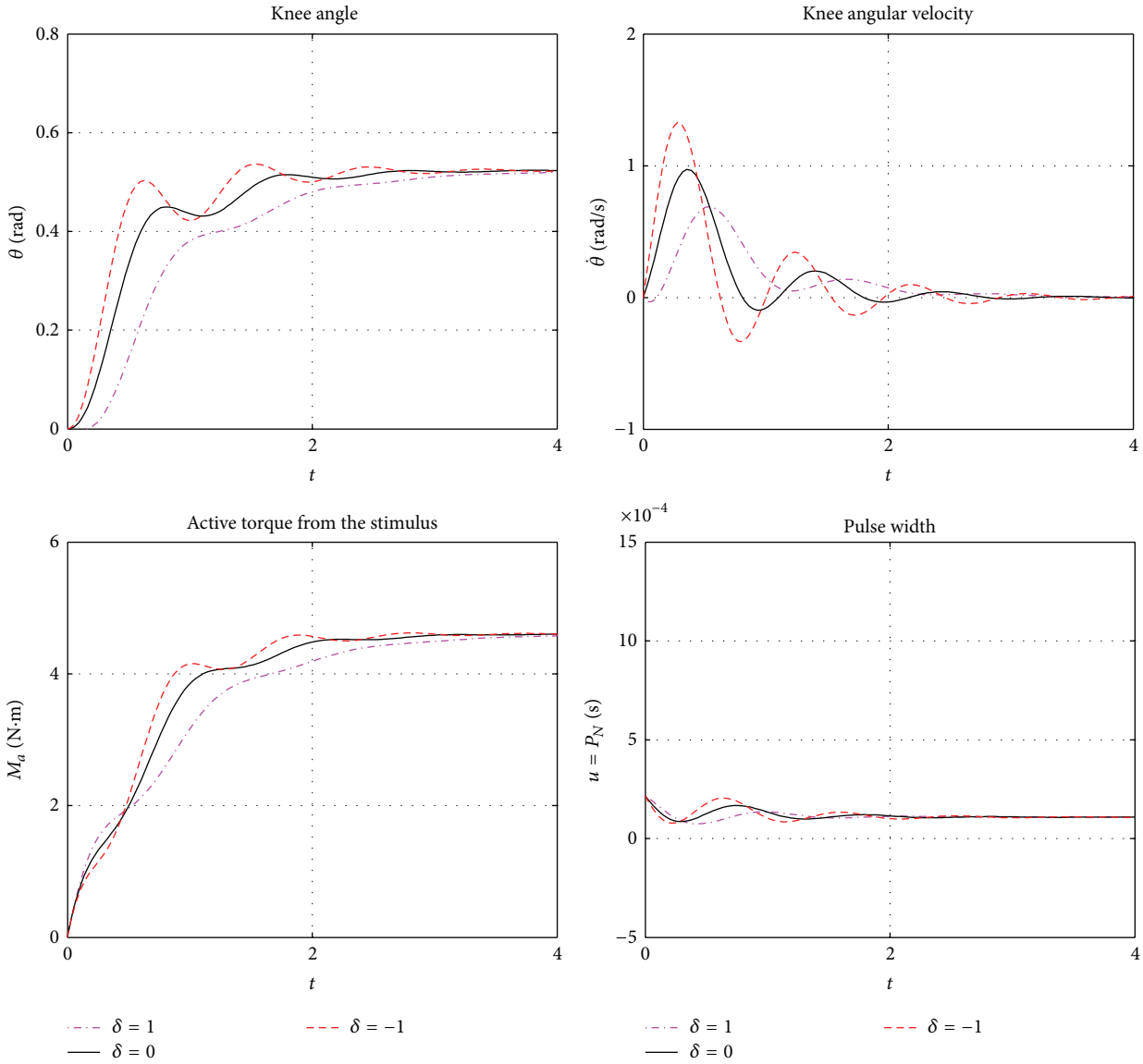


FIGURE 3: Simulation of the dynamic equation (10) of the paraplegic model for the operating point of 30°, $f_{21\,\max} = -21.7834$ and $f_{21\,\min} = -36.0086$, for $\delta = 0$, $\delta = 1$, and $\delta = -1$.

4. Results

4.1. Results without considering Output Constraints. Considering the parameters defined in [5, 7] (see Table 1) for $\theta_0 = 30^\circ$ and $-30^\circ \le x_1 \le 30^\circ$, it follows that

$$x(0) = \begin{bmatrix} -\theta_{v0} & 0 & -M_{a0} \end{bmatrix}^T = \begin{bmatrix} -0.5236 & 0 & -4.6068 \end{bmatrix}^T,$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -28.8960 & -0.7459 & 2.7624 \\ 0 & 0 & -1.0515 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 44690 \end{bmatrix}^T, \tag{21}$$

$$L = \begin{bmatrix} 0 & 7.1126 & 0 \end{bmatrix}^T,$$

$$C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

Applying the LMIs (11) and (12), using the *software* Matlab [18], for $\theta_0 = 30^\circ$, the state feedback matrix obtained was

$$K = \begin{bmatrix} -0.3244 & 0.0999 & 0.0605 \end{bmatrix} \times 10^{-3} \tag{22}$$

and the constant ϵ was

$$\epsilon = 12.7728. \tag{23}$$

In Figure 3, the outputs of dynamic equation's simulations show that the graphics of the knee angles start at 0°, corresponding to $\Delta x_1 = \theta_{v0} = \pi/6 = -0.5236$ (rad). The graphics of the angular velocity of the knee start up at 0 (rad/s). Finally, the graphics of the active torques produced by the stimuli are initiated at 0 (N · m) corresponding to $\Delta x_3 = M_{a0} = 4.6068$ (see Section 2). Besides, the transient response of the dynamic compensator simulation applied to the

Table 2: Poles of closed loop (10) for $\theta_0 = 30^\circ$, for $\delta = 0$, $\delta = 1$, and $\delta = -1$.

$\delta = 0$	$\delta = 1$	$\delta = -1$
$-1.3777 \pm j6.1086$	$-1.6143 \pm j5.4951$	$-1.2186 \pm j6.6799$
-1.7463	-1.2732	-2.0646

dynamic of the paraplegic patient, with $f_{21\max} = -21.7834$, $f_{21\min} = -36.0086$, and the initial state is shown. Cases can be studied with other operation points, as shown in [2, 11], where a linearization result was considered for each operation point: $\pi/4$ (rad) and $\pi/3$ (rad).

The closed loop poles of system (10) are given in Table 2. In the first column of the table there are closed loop poles of the nominal system and in this case $\delta = 0$. In the other two columns of the table there are closed loop poles considering the extreme values of the $\widetilde{f}_{21}(x_1)$, respectively, for $\delta = 1$ and $\delta = -1$. In all table columns it is possible to verify that the real parts of the poles are negative, which means that they are positioned in the left upper half-plane of the s-plane.

In Figure 3, the knee angle graphic for $\delta = 0$, the curve stabilizes at $\pi/6 = 0.52$ (rad) in a time of about 3.5 (s). The graphic of active torque produced by the electric stimulus shows that the curve stabilizes at the mark of 4.6 (N · m) at 3.1 (s). The graph of the angular velocity of the knee stabilizes in the mark of 0 (rad/s) at 3.2 (s). The pulse width graphic stabilizes in 1.07×10^{-4} at about 2.0 (s).

4.2. Results considering Output Constraints. In the second simulation, applying the LMIs (11), (12), and (14), with $\lambda_0 = 6.8018$, using the *software* Matlab [18], for $\theta_0 = 30^\circ$, the state feedback matrix obtained was

$$K = \begin{bmatrix} -0.2639 & 0.0602 & 0.0305 \end{bmatrix} \times 10^{-3} \tag{24}$$

and the constant ϵ was

$$\epsilon = 4.2263. \tag{25}$$

The closed loop poles of the system are given in Table 3. In the first column of the table there are the closed loop poles of the nominal system and in this case $\delta = 0$. In the other two columns of the table there are the closed loop poles considering the extreme values of the $\widetilde{f}_{21}(x_1)$, respectively, for $\delta = 1$ and $\delta = -1$. In all table columns it is possible to verify that the real parts of the poles are negative, which means that they are positioned in the left upper half-plane of the s-plane.

In Figure 4, the knee angle graphic for $\delta = 0$, the curve stabilizes at $\pi/6 = 0.52$ (rad) in a time of about 4.0 (s). The graphic of active torque produced by the electric stimulus shows that the curve stabilizes at the mark of 4.6 (N · m) at 4.0 (s). The graph of the angular velocity of the knee stabilizes in the mark of 0 (rad/s) at 4.0 (s). The pulse width graphic stabilizes in 1.07×10^{-4} at about 2.7 (s).

In both designs, no decay rate was specified.

Equation (5) shows that it is necessary that the pulse width satisfies $P_N > -M_{a0}/G$. Actually, any value of P_N greater than the ratio $-M_{a0}/G$ satisfies the requirement. However, the previous relation implies that P_N between $-M_{a0}/G$ and M_{a0}/G satisfies the requirement (see Figure 4).

Table 3: Poles of closed loop (10) for $\theta_0 = 30^\circ$, $\lambda_0 = 6.8018$, for $\delta = 0$, $\delta = 1$, and $\delta = -1$.

$\delta = 0$	$\delta = 1$	$\delta = -1$
$-1.0627 \pm j5.8995$	$-1.2390 \pm j5.2721$	$-0.9457 \pm j6.4770$
-1.0343	-0.6818	-1.2681

5. Discussions

As can be seen, the plant input is the functional electrical stimulation pulse applied on the quadriceps muscle. The plant output is the movement of the leg of the paraplegic patient until an angle allowed by the conditions of the patient. So the initial objective was achieved: a state feedback matrix, K, was obtained considering norm-bounded uncertainties, such that the system in question becomes stable. The designed controller is able to make the patient's leg move from the rest to a desired angle and once the stimulation of the controller is removed, the leg goes back to the rest situation by gravity.

It was proved that Theorem 1 ensures stability for the open-loop system, while Theorem 2 ensures the asymptotic stability for the closed loop system. This fact can be confirmed by the analysis of Table 3 where it can be seen that the poles of the plant for different operating points have negative real parts, which means that they are positioned in the upper left half-plane of the s-plane.

Along with the study of the stability, graphical results could have been generated to control the speed of transient response, specifying a decay rate. It allows a shorter time recovery, as shown in [2].

As it is clear, a negative pulse width is impractical. Then restrictions must be imposed on input, as shown in Figure 5, to prevent the pulse width from reaching zero or negative values. According to Gaino in [2] with an appropriate combination of the decay rate and the upper limiting value for the variation of pulse width one can achieve the increasing or the decreasing of the control law amplitude.

The resulting simulation outputs shown in Figure 4 are identical to that obtained by Gaino in [2] in the same conditions. Here the simulation results were obtained using a restriction imposed on the output by a factor $\lambda_0 = 6.8018$, as it can be verified in Table 3. As a consequence, the oscillation was eliminated from the pulse width curve, stabilizing in values of P_N very close to one another in both simulations.

As mentioned by Sanches et al. in [8] "An electric current with a balanced, biphasic, rectangular waveform with a pulse width of 220 μs was applied to the skin surface of the volunteer. The signal amplitude was then held constant for 3 seconds and considered a "step pattern" [5]. The pulse amplitude was adjusted to obtain an incomplete knee extension, reaching a maximum angle of 70° with respect to the initial rest position. The frequency of the signal was 50 Hz [5]. For healthy persons, currents with amplitudes in the range of 60–80 mA are usually used, but in paraplegic patients, currents can reach 120 mA." As mentioned by Ferrarin and Pedotti in [5], "In this way the presence of nonlinearity due to the muscle recruitment threshold and knee joint end stop could

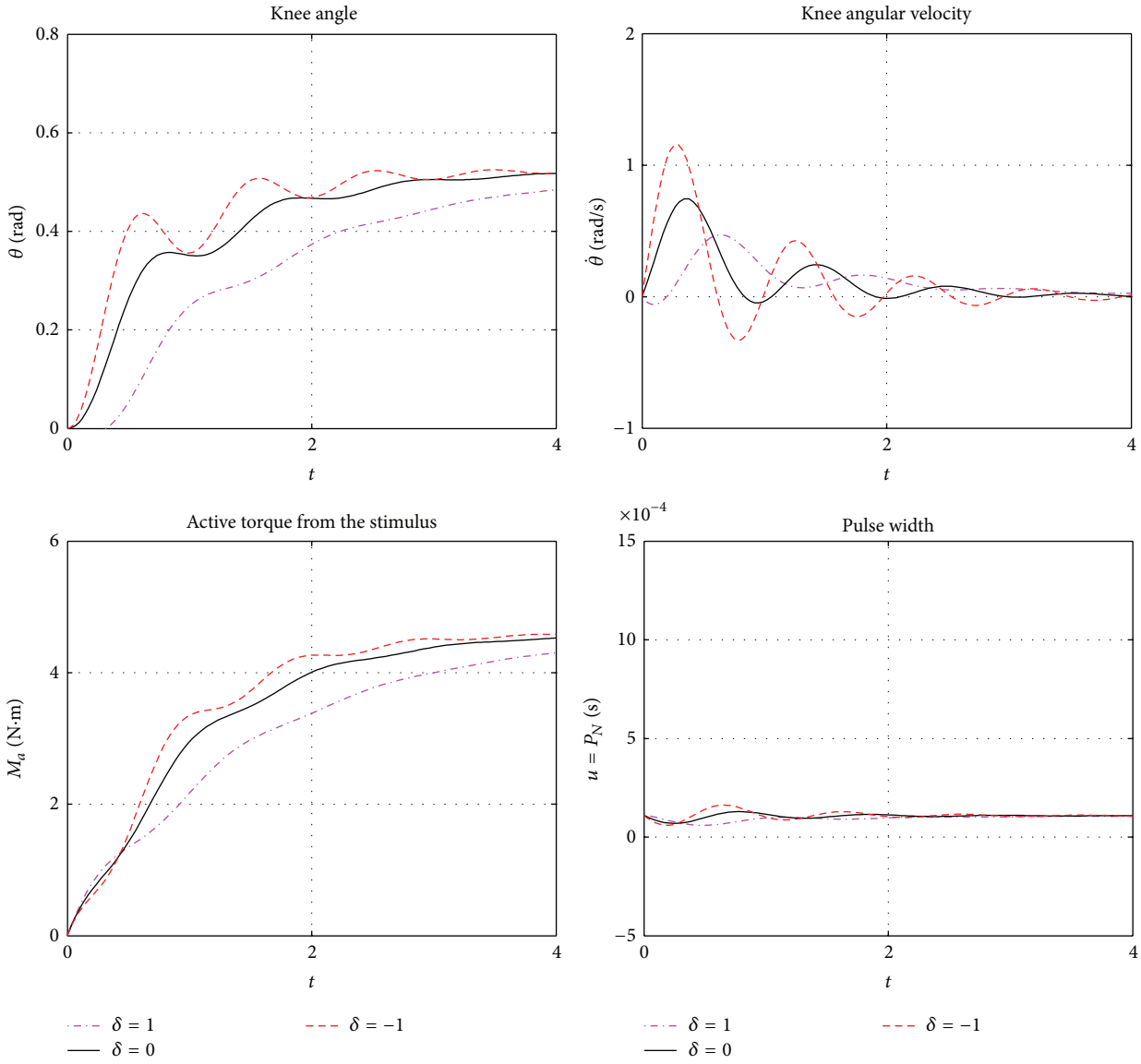


FIGURE 4: Simulation of the dynamic equation (10) of the paraplegic model for the operating point of 30°, $f_{21\,\max} = -21.7834$ and $f_{21\,\min} = -36.0086$, using bounded output by the factor $\lambda_0 = 6.8018$, for $\delta = 0$, $\delta = 1$, and $\delta = -1$.

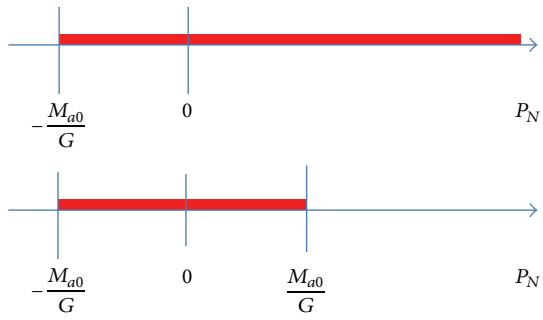


FIGURE 5: Range restriction for the pulse width.

be excluded." It is less dangerous to apply step patterns than sinusoidal wave forms.

6. Conclusions

The initial objective was achieved since a state feedback matrix, K, considering norm-bounded uncertainties, was obtained such that the system in question became stable. The designed controller is able to make the patient's leg move from the rest to a desired angle and once the stimulation is removed, the leg goes back to the rest situation by gravity.

It was proved that Theorem 1 ensures stability for the open-loop system, while Theorem 2 ensures the asymptotic stability for the closed loop system. The simulations of Section 4 show that the curves for the cases where $\delta = 0$ are situated between the curves corresponding to the extremes values of δ, that is, $\delta = 1$ and $\delta = -1$, as it was expected. Table 3 shows that, for all values of δ, the real parts of the poles are negative, and this indicates that the controller designed is able to maintain the stability of the plant.

Conflict of Interests

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

Acknowledgments

The authors gratefully acknowledge the Brazilian agencies CAPES, CNPq, and FAPESP, for the financial support.

References

[1] R. L. Lieber, P. D. Silva, and D. M. Daniel, "Equal effectiveness of electrical and volitional strength training for quadriceps femoris muscles after anterior cruciate ligament surgery," *Journal of Orthopaedic Research*, vol. 14, no. 1, pp. 131–138, 1996.

[2] R. Gaino, *Controle de movimentos de pacientes paraplégicos utilizando modelos Fuzzy T-S, Tese (Doutorado)—Programa de Pós-Gradua ção em Engenharia Elétrica*, Universidade Estadual Paulista "Júlio de Mesquita Filho", Ilha Solteira, São Paulo, Brazil, 2009, (Portuguese).

[3] H. E. Makssoud, D. Guiraud, and P. Poignet, "Mathematical muscle model for functional electrical stimulation control strategies," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1282–1287, New Orleans, La, USA, 2004.

[4] S. J. Dorgan and M. J. O'Malley, "A mathematical model for skeletal muscle activated by N-let pulse trains," *IEEE Transactions on Rehabilitation Engineering*, vol. 6, no. 3, pp. 286–299, 1998.

[5] M. Ferrarin and A. Pedotti, "The relationship between electrical stimulus and joint torque: a dynamic model," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 3, pp. 342–352, 2000.

[6] R. Gaino, M. C. M. Teixeira, A. A. de Carvalho et al., "Realimentação derivativa e modelo fuzzy Takagi-Sugeno para controle da articulação do joelho de pacientes paraplégicos com o uso de acelerômetros," *Brazilian Journal of Biomedical Engineering*, vol. 27, no. 2, pp. 67–78, 2011 (Portuguese).

[7] M. C. M. Teixeira, G. S. Deaecto, R. Gaino, E. Assunção, A. A. de Carvalho, and U. C. Farias, "Design of a fuzzy Takagi-Sugeno controller to vary the joint knee angle of paraplegic patients," in *Neural Information Processing*, vol. 4234 of *Lecture Notes in Computer Science*, pp. 118–126, Springer, Berlin, Germany, 2006.

[8] M. A. A. Sanches, R. Gaino, R. F. Kozan et al., "Digital controllers desing considering hardware constraints: application in a paraplegic patient," *Brazilian Journal of Biomedical Engineering*, vol. 30, no. 3, pp. 232–241, 2014.

[9] J. Cao, "Improved delay-dependent stability conditions for MIMO networked control systems with nonlinear perturbations," *The Scientific World Journal*, vol. 2014, Article ID 196927, 4 pages, 2014.

[10] B. J. Rosa Filho, "Biomecânica Global," 2013, http://www.wgate.com.br/conteudo/medicinaesaude/fisioterapia/biomecanica.htm.

[11] D. N. T. P. Diogo, L. H. R. Bueno, M. R. Covacic, and R. Gaino, "Modelo Fuzzy Takagi-Sugeno para controle do ângulo de articulação do joelho de pacientes paraplégicos," *Semina: Ciências Exatas e Tecnológicas*, vol. 32, no. 2, pp. 215–228, 2012 (Portuguese).

[12] R. Gaino, M. C. M. Teixeira, R. Cardim, A. A. de Carvalho, E. Assunção, and M. A. A. Sanches, "Controle não-linear Takagi-Sugeno do movimento de paraplégicos utilizando acelerômetros," in *XXI Congresso Brasileiro de Engenharia Biomédica*, pp. 1254–1257, CBEB, Salvador, Brazil, (Portuguese).

[13] R. Lu, W. Du, J. Wang, and A. Xue, "Robust H_2 control for descriptor system based on new bounded real lemma," *Circuits, Systems, and Signal Processing*, vol. 28, no. 6, pp. 869–882, 2009.

[14] S. P. Boyd, L. El-Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 1994.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, New York, NY, 7th edition, 2009.

[16] M. R. Covacic, M. C. M. Teixeira, E. Assunção, and R. Gaino, "LMI-based algorithm for strictly positive real systems with static output feedback," *Systems & Control Letters*, vol. 61, no. 4, pp. 521–527, 2012.

[17] L. A. V. R. Filho, *Filtragem Ótima Robusta em Sistemas Dinâmicos [Dissertação de Mestrado]*, Faculdade de Engenharia Elétrica e Engenharia de Computação, Universidade Estadual de Campinas, São Paulo, Brazil, 2004, (Portuguese).

[18] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox User's Guide*, The Mathworks Inc., Natick, Mass, USA, 1995.