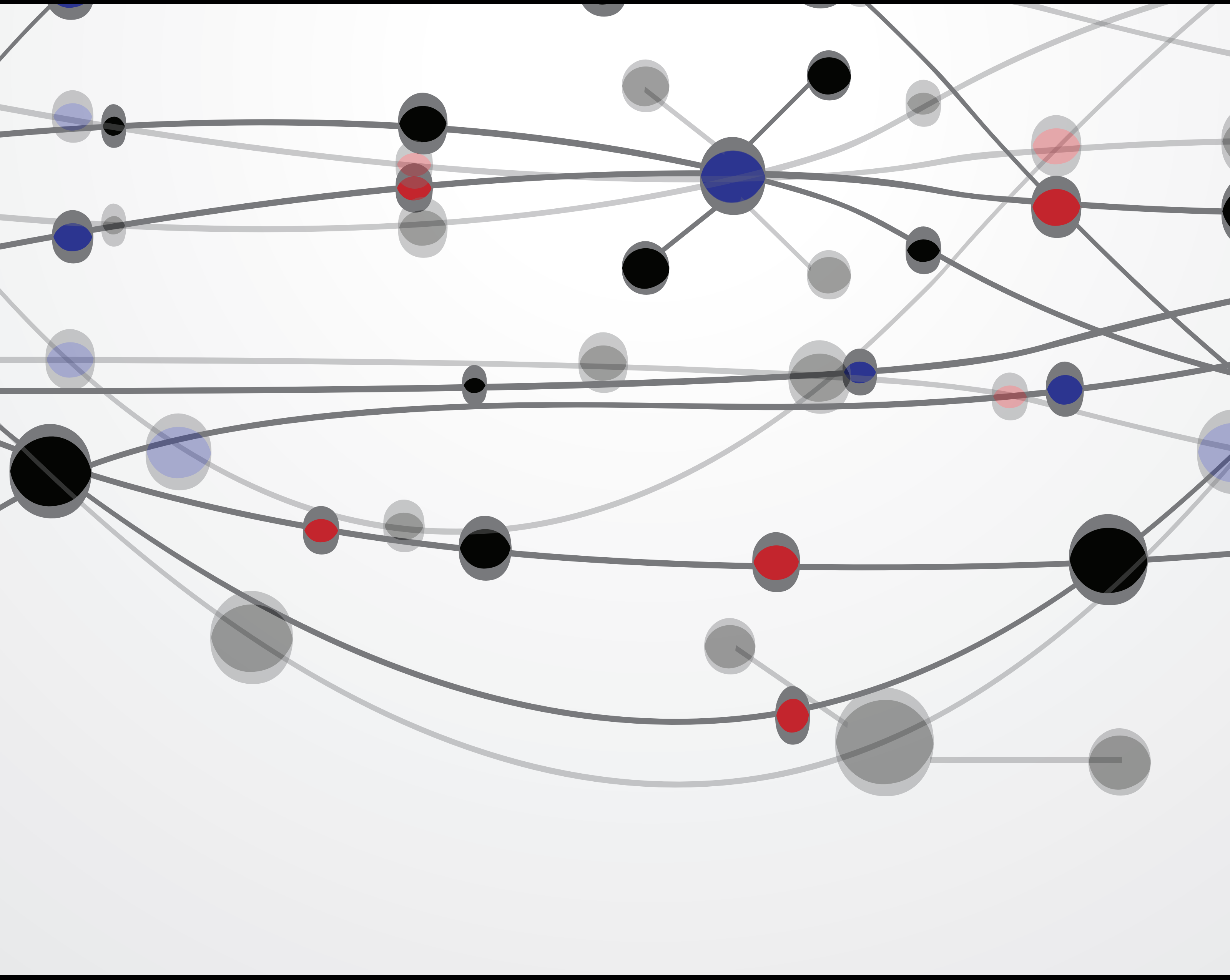
The Scientific World Journal

# Bioinspired Computation and Its Applications in Operation Management

Guest Editors: Tinggui Chen, Jianjun Yang, Kai Huang, and Qiang Cheng



# Bioinspired Computation and Its Applications in Operation Management

The Scientific World Journal

# Bioinspired Computation and Its Applications in Operation Management

Guest Editors: Tinggui Chen, Jianjun Yang, Kai Huang, and Qiang Cheng

# Contents

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 356571, 5 pages
http://dx.doi.org/10.1155/2014/356571

*Editorial*

# Bioinspired Computation and Its Applications in Operation Management

**Tinggui Chen,[1] Jianjun Yang,[2] Kai Huang,[3] and Qiang Cheng[4]**

[1] *School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[2] *Department of Computer Science, University of North Georgia, Oakwood, GA 30566, USA*
[3] *DeGroote School of Business, McMaster University, DSB 404, Hamilton, ON, Canada L8S 4M4*
[4] *College of Mechanical Engineering and Applied Electronics, Beijing University of Technology, Beijing 100124, China*

Correspondence should be addressed to Tinggui Chen; ctgsimon@mail.zjgsu.edu.cn

Received 23 March 2014; Accepted 23 March 2014; Published 12 June 2014

Bioinspired computation is an umbrella term for different computational technologies that are based on principles or models of biological systems. This class of approaches, including evolutionary algorithm, swarm intelligence, and artificial immune system, complements traditional ones in the sense that the former can be applied to large and complex combination optimization problems, but the latter encounters difficulties. Therefore, bioinspired technologies are becoming important in the face of solving discrete and dynamic problems.

Recently, the bioinspired computation has attracted much attention of researchers and has also been widely applied to operation management fields ranging between production assembling, inventory control, project scheduling, human resource management, and revenue management. However, due to complexity and uncertainty in operation management problems, it is very difficult to find out the optimum solution under the limited resources, time, and money in real-world applications using the bioinspired technologies. Therefore, it is necessary to develop efficient or improved algorithms to solve operation management problems.

The main objective of this special issue is to present the original research and review articles on the latest theoretical and practical achievements that will contribute to the field of bioinspired computation and its applications in operation management, in all branches of management science and computer science.

The special issue received 66 high-quality submissions from different countries all over the world. All submitted manuscripts have followed the same standard (peer-reviewed by at least three independent reviewers) as applied to regular ones to "this journal." Due to space limit, only 24 papers could be published (acceptance ratio of 1 : 3). Inevitably, difficult decisions had to be made, and some high-quality submissions could not be included. The primary guideline was to demonstrate the wide scope of bioinspired computation and applications in operation management. Besides, some novel research questions from different applications that are worth further investigation in the future are also included.

In the paper, "*A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under e-supply chain environment*," Y. Li et al. formulate a location-inventory-routing problem model with no quality defects' returns. In addition, to solve this NP-hard problem, an effective hybrid genetic-simulated annealing algorithm (HGSAA) is proposed. Results of numerical examples show that HGSAA outperforms GA in computing time, optimal solution, and computing stability. Moreover, the proposed model is also very useful in helping managers make the right decisions under e-supply chain environment.

In the paper "*Optimization and planning of emergency evacuation routes considering traffic control*," G. Li et al. establish two different emergency evaluation models on the basis of the maximum-flow model (MFM) and minimum-cost maximum-flow model (MC-MFM) and provide the corresponding algorithms for the evacuation of one source node to one designated destination (one-to-one evacuation). Besides, they also extend their model to solve one-to-many evacuations. Finally, case analysis of evacuation optimization and planning in Beijing is given and the efficiency of the proposed model is illustrated.

In the paper "*An improved hierarchical genetic algorithm for sheet cutting scheduling with process constraints,*" Y. Rao et al. proposed an improved hierarchical genetic algorithm for sheet cutting problem which involves $n$ cutting patterns for $m$ nonidentical parallel machines with process constraints. Furthermore, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and a kind of mutation probability are used in this algorithm. The computational result and comparison prove that the presented approach is quite effective for the considered problem.

In the paper entitled "*An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization*," X. Yu et al. formulate a novel adaptive hybrid algorithm based on PSO and DE (HPSO-DE) by developing a balanced parameter between PSO and DE. Adaptive mutation is carried out to current population when the population clusters around local optima. The HPSO-DE enjoys the advantages of PSO and DE and maintains diversity of the population. Compared with PSO, DE, and their variants, the performance of HPSO-DE is promising and competitive.

In the paper "*An effective hybrid self-adapting differential evolution algorithm for the joint replenishment and location-inventory problem in a three-level supply chain*," L. Wang et al. provide an effective intelligent algorithm for a modified joint replenishment and location-inventory problem (JR-LIP) where distribution centers (DCs) replenish their demands jointly. The problem of the JR-LIP is to determine the number and locations of DCs to be opened, the assignment of retailers to DCs, the basic replenishment time of DCs, and the replenishment frequency of each DC such that the overall cost is minimized. To find an effective approach for the JR-LIP, a hybrid self-adapting differential evolution algorithm (HSDE) is designed. Comparative results of benchmark functions' tests and randomly generated JR-LIPs show that HSDE outperforms GA and HDE.

In the paper entitled "*Genetic algorithm application in optimization of wireless sensor networks*," A. Norouzi and A. H. Zaim use genetic algorithm to optimize wireless sensor networks. A fitness function with optimum formula was obtained and the present protocols are optimized. The results of simulations in JPAC, MATLAB, and NS are compared with that of the present protocols and optimization of the two parameters is confirmed. It is also noticeable that the diagrams obtained from the simulations show an improvement in energy consumption parameters and lifetime of the network; this means more ideal WSNs.

In the paper, the research of X. Cheng and Y. Lin entitled "*Multiobjective robust design of the double wishbone suspension system based on particle swarm optimization*" proposes a robust design based on bioinspired computation. The simulation experiment is arranged and Latin hypercube design is adopted to find the initial point. Then sensitivity analysis is utilized to determine main design variables. The kriging model is employed for fitting the mean and variance of the quality characteristics according to the simulation results. Furthermore, a particle swarm optimization method based on simple PSO is applied and the tradeoff between the mean and deviation of performance is made to solve the robust optimization problem of double wishbone suspension system.

In the paper entitled "*A novel artificial bee colony approach of live virtual machine migration policy using Bayes theorem*," G. Xu et al. present a novel heuristic approach which is called PS-ABC. Its algorithm includes two parts. One is that it combines the ABC (artificial bee colony) idea with the uniform random initialization idea and the binary search idea and Boltzmann selection policy to achieve an improved ABC-based approach with better global exploration's ability and local exploitation's ability. The other one is that it uses the Bayes Theorem to further optimize the improved ABC-based process to faster get the final optimal solution. As a result, the whole approach achieves a longer-term efficient optimization for power saving. The experimental results demonstrate that PS-ABC evidently reduces the total incremental power consumption and better protects the performance of VM running and migrating compared with the existing research. It makes the result of live VM migration more high effective and meaningful.

In the paper entitled "*Seven-spot ladybird optimization: a novel and efficient metaheuristic algorithm for numerical optimization*," P. Wang et al. present a novel biologically inspired metaheuristic algorithm called seven-spot ladybird optimization (SLO). The SLO is inspired by recent discoveries on the foraging behavior of a seven-spot ladybird. In this paper, the performance of the SLO is compared with that of the genetic algorithm, particle swarm optimization, and artificial bee colony algorithms by using five numerical benchmark functions with multimodality. The results show that SLO has the ability to find the best solution with a comparative small population size and is suitable for solving optimization problems with lower dimensions.

In the paper entitled "*Pricing resources in LTE networks through multiobjective optimization,*" Y.-L. Lai and J.-R. Jiang study the pricing resources with profits and satisfaction optimization (PRPSO) problem in the LTE networks, considering the operator profit, and subscribe satisfaction at the same time. The problem is modeled as nonlinear multiobjective optimization with two optimal objectives: (1) maximizing operator profit and (2) maximizing user satisfaction. They propose solving the problem based on the framework of the NSGA-II algorithm. Simulations are conducted for evaluating the proposed solution.

In the paper entitled "*Comparison of multiobjective evolutionary algorithms for operations scheduling under machine availability constraints*," M. Frutos et al. analyze different evolutionary multiobjective algorithms (MOEAs) for this kind of problems. They consider an experimental framework in which they schedule production operations for four real-world Job-Shop contexts using three algorithms, NSGAII, SPEA2, and IBEA. Using two performance indexes, hypervolume and R2, they found that SPEA2 and IBEA are the most efficient for the tasks at hand. On the other hand IBEA seems to be a better choice of tool since it yields more solutions in the approximate Pareto frontier.

In the paper entitled "*A modified decision tree algorithm based on genetic algorithm for mobile user classification problem*," D.-S. Liu and S.-J. Fan put forward a modified decision tree algorithm for mobile user classification, which introduced genetic algorithm to optimize the results of the decision tree algorithm. They also take the context information as a classification attributes for the mobile user and classify the context into public context and private context classes. Then, the processes and operators of the algorithm are analyzed. At last, an experiment is given so as to verify the efficiency and effectiveness of the proposed approach.

In the paper "*A location selection policy of live virtual machine migration for power saving and load balancing*," J. Zhao et al. present the specific design and implementation of MOGA-LS such as the design of the genetic operators and fitness values and elitism. They introduce the Pareto dominance theory and the SA (simulated annealing) idea into MOGA-LS and present the specific process to get the final solution. And thus the whole approach achieves a long-term efficient optimization for power savings and load balancing. The experimental results demonstrate that MOGA-LS evidently reduces the total incremental power consumption, better protects the performance of VM migration, achieves the balancing of system load compared with the existing research. It makes the result of live VM migration more high effective and meaningful.

In the paper "*Modeling Markov switching ARMA-GARCH neural networks models and an application to forecasting stock returns,*" M. E. Bildirici and O. Ersin propose a family of regime switching GARCH neural network models to model volatility. Proposed MS-ARMA-GARCH-NN models allow MS type regime switching in both the conditional mean and conditional variance for time series and further augmented with artificial neural networks to achieve improvement in forecasting capabilities. In the empirical section, daily stock returns in ISE100 Istanbul Stock Index are modeled and forecasted. Forecast success is evaluated with MAE, MSE, and RMSE criteria and Diebold-Mariano tests. Result suggest that hybrid MLP and time lag recurrent MS-ARMA-FIAPGARCH hybrid MLP and MS-ARMA-FIAPGARCH-RNN provided the best forecast and modeling performance over the simple GARCH models and additionally Gray's MS-GARCH model. Therefore, augmenting the forecasting capabilities with neural networks and regime switching provide significant gains in various economic applications.

In the paper entitled "*Full glowworm swarm optimization algorithm for whole-set orders scheduling in single machine*," Z. Yu and X. Yang propose a new glowworm swarm optimization algorithm for scheduling by analyzing the characteristics of whole-set orders problem and combining the theory of glowworm swarm optimization. A new hybrid-encoding schema combining with two-dimensional encoding and random-key encoding is given. In order to enhance the capability of optimal searching and speed up the convergence rate, the dynamical changed step strategy is integrated into this algorithm. Furthermore, experimental results prove its feasibility and efficiency.

In the paper "*Multiple R&D scheduling optimization with improved particle swarm algorithm*," M. Liu et al. discuss the features of multiple R&D environment in customization enterprises and demands of resources distribution, make some improvements to the multiple project scheduling models, and put forward a multiple project crashing scheduling model based on postpone-punishment in large scale customization environment. At the same time, based on the analysis of particle swarm optimization algorithm and its improvement, a new solution of dynamic center particle swarm optimization algorithm to multiproject scheduling model has been arisen. The best solution can be gained after the experiment to the illustration example through MATLAB program; thus, the model in the paper and application of algorithm can be proved.

In the paper "*Applying probability theory for the quality assessment of a wildfire spread prediction framework based on genetic algorithms*," A. Cencerrado et al. present a framework for assessing how the existing constraints at the time of attending an ongoing forest fire affect simulation results, both in terms of quality (accuracy) obtained and the time needed to make a decision. The core of this framework is evaluated according to the probability theory principles. Thus, a strong statistical study is presented, oriented towards the characterization of such an adjustment technique in order to help the operation managers to deal with the two aspects previously mentioned: time and quality. The experimental work in this paper is based on a region in Spain which is one of the most prone to forest fires: El Cap de Creus.

In the paper "*Scheduling projects with multiskill learning effect*," H. Zha and L. Zhang investigate the project scheduling problem with multiskill learning effect. A new model is proposed to deal with the problem, where both autonomous and induced learning are considered. In order to obtain the optimal solution, a genetic algorithm with specific encoding and decoding schemes is introduced. A numerical example is used to illustrate the proposed model. The computational results show that the learning effect cannot be neglected in project scheduling. By means of determining the level of induced learning, the project manager can balance the project makespan with total cost.

In the paper entitled "*A novel algorithm combining the finite state method and genetic algorithm for scheduling of crude oil problem*," Q.-Q. Duan et al. propose a novel genetic algorithm to solve the MINLP problem. The MINLP model they discussed is based on the single-operation sequencing (SOS) time representation. In this paper, based on the sequencing rules and the extension to the regular expression calculus, a deterministic finite state automaton (DFA) which captures valid possible schedule sequences is constructed. In the initial population stage of GA, a population of candidate solutions is generated on the basis of this DFA. The rule-based mutation strategy consists in following the sequencing rule and meeting the nonoverlapping constraint, which is moving towards optimal solutions efficiently. The optimization results indicate both the effectiveness of the model and efficiency as well as the robustness of the solution methodology.

In the paper "*Improved particle swarm optimization with a collective local unimodal search for continuous optimization problems*," M. A. Arasomwan and A. O. Adewumi propose a new local search technique and use it to improve the performance of particle swarm optimization algorithms by addressing the problem of premature convergence. In the proposed local search technique, a potential particle position in the solution search space is collectively constructed by a number of randomly selected particles in the swarm. The number of times the selection is made varies with the dimension of the optimization problem and each selected particle donates the value in the location of its randomly selected dimension from its personal best. After constructing the potential particle position, some local search is done around its neighborhood in comparison with the current swarm global best position. It is then used to replace the global best particle position if it is found to be better; otherwise, no replacement is made. Using some well-studied benchmark problems with low and high dimensions, numerical simulations are used to validate the performance of the improved algorithms. Comparisons are made with four different PSO variants; two variants implement different local search technique, while the other two do not. Results show that the improved algorithms could obtain better quality solution while demonstrating better convergence velocity and precision, stability, robustness, and global-local search ability than the competing variants.

In the paper "*A new collaborative recommendation approach based on users clustering using artificial bee colony algorithm*," C. Ju and C. Xu propose a novel collaborative filtering recommendation approach based on $K$-means clustering algorithm. Firstly, they use artificial bee colony (ABC) algorithm to overcome $K$-means algorithm's problems. And then they adopt the modified cosine similarity considering products' popularity degrees and users' preference degrees to compute the similarity between users in the same clusters. Finally, they generate the recommendation results for target users. Detailed numerical analysis on a benchmark dataset *MovieLens* and a real-world dataset indicates that their new collaborative filtering approach based on users clustering algorithm outperforms many other recommendation methods.

In the paper "*Comprehensive optimization of emergency evacuation route and departure time under traffic control,*" G. Li et al. investigate the comprehensive optimization of major emergency evacuation route and departure time, in which evacuation propagation mechanism is considered under traffic control. Based on practical assumptions, they first establish a comprehensive optimization model based on the simulation of evacuation route and departure time. In order to optimize the evacuation routes and departure time, they explore the reasonable description methods of evacuation traffic flow propagation under traffic control, including the establishment of traffic flow propagation model and the design of algorithm which can implement simulation of evacuation traffic flow. Finally, they propose a heuristic algorithm for optimization of this comprehensive model. In case analysis, they take some areas in Beijing as evaluation sources to verify the reliability of this model. Moreover, some constructive suggestions for Beijing's emergency evacuation are proposed, which can be applied to actual situation, especially under traffic control.

In the paper "*Collaborative scheduling model for Supply-Hub with multiple suppliers and multiple manufacturers*," G. Li et al. investigate a collaborative scheduling model that contains multiple suppliers, multiple manufacturers, and a Supply-Hub. They describe the operational process of Supply-Hub and formulate the basic scheduling model. Based on this, they consider two different scenarios: one is that the suppliers and the manufacturers make their decisions separately and the other is that Supply-Hub makes the entire decisions with the collaborative scheduling. Under this condition, they prove that the scheduling model with Supply-Hub is NP-complete issue; thus, an autoadapted differential evolution algorithm is proposed. In the numerical analysis, they illustrate that the performance of collaborative scheduling for Supply-Hub is superior to separate decision made by each manufacturer and supplier. Furthermore, they also show that the proposed algorithm has a good convergence and reliability, in particular when it can be exerted flexibly onto certain suppliers.

In the paper entitled "*Hierarchical artificial bee colony algorithm for RFID network planning optimization*," L. Ma et al. present an optimization model for planning the positions and radiated power setting of readers in the RFID network. The four nonlinear RNP objective functions are formulated with considering tag coverage, reader interference, economic efficiency, and network load balance as the primary requirements of the real-world RFID system. And the combined measure is also given so that the multiple objectives can be optimized simultaneously. Finally, in order to solve the RNP model, a novel hierarchical artificial colony algorithm, called HABC, is proposed by extending single artificial bee colony (ABC) algorithm to hierarchical and cooperative mode by combining the multipopulation cooperative coevolution approach based on vector decomposing strategy and the comprehensive learning method. Results obtained from the proposed approach have been compared with those obtained by ABC, PSO, CPSO, EGA, CMA_ES, and CCEA. The experiment results show that, for all the test functions, the HABC gets significant superiority to other six algorithms. HABC is then employed to solve the real-world RNP problem on two different-scale instances, namely, Cd100 and Rd500. Through simulation studies, the HABC remarkably outperforms other algorithms. Especially, in tackling larger-scale RNP problem (i.e., Rd500 instance), the HABC performs more effectively, indicating that the HABC is more suitable for solving high-dimension RNP problem.

The study of bioinspired computation and its applications in operation management is still in its early stage. This special issue demonstrates the theoretical and practical importance of further studies on bioinspired computation.

## Acknowledgments

We would like to express our gratitude to all of the authors for their contribution and the reviewers for their effort in

providing valuable comments and feedback. We hope that this special issue offers a comprehensive and timely view of the area of applications of bioinspired computation and that it will offer stimulation for further research.

*Tinggui Chen*
*Jianjun Yang*
*Kai Huang*
*Qiang Cheng*

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 652135, 7 pages
http://dx.doi.org/10.1155/2014/652135

*Research Article*

# Multiple R&D Projects Scheduling Optimization with Improved Particle Swarm Algorithm

**Mengqi Liu, Miyuan Shan, and Juan Wu**

*School of Business Administration, Hunan University, Changsha, Hunan 410082, China*

Correspondence should be addressed to Mengqi Liu; liumengqi1976@163.com

Received 29 December 2013; Accepted 10 March 2014; Published 12 June 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

For most enterprises, in order to win the initiative in the fierce competition of market, a key step is to improve their R&D ability to meet the various demands of customers more timely and less costly. This paper discusses the features of multiple R&D environments in large make-to-order enterprises under constrained human resource and budget, and puts forward a multi-project scheduling model during a certain period. Furthermore, we make some improvements to existed particle swarm algorithm and apply the one developed here to the resource-constrained multi-project scheduling model for a simulation experiment. Simultaneously, the feasibility of model and the validity of algorithm are proved in the experiment.

## 1. Introduction

As we all know, large make-to-order enterprises can meet customers' special demands with quick response and relatively low cost. The process is usually composed of three stages: order R&D, sample manufacturing, and batch manufacturing. Whether the profits can be maximized and the requirements of customers on quality and delivery can be met are usually determined at the order R&D stage, during which time the manufacturing ability and the cost composition of the order are informed, and this stage occupies 3/4 of the time needed to finish the order. So it is no doubt that the key competitiveness of large make-to-order enterprises is the order R&D ability. As a result, improving the order R&D ability as well as reducing the cost and shortening the delivery time at R&D stage is crucial to large make-to-order enterprises.

Resource-constrained multi-project scheduling problem is a generalization of the resource-constrained project scheduling problem (RCPSP). And the features of multiple R&D projects in large make-to-order enterprises are as follows: (1) results and timing are very uncertain because the unique characteristic of each project is based upon the degree of innovation; (2) human resource is the main and scarcest resource in R&D projects; (3) the cost of each project is supposed to be as low as possible owing to the limited budget and for the pursuit of expected profits; (4) multi-project R&D environments are dynamic, which originate from technological updating as well as continuous new orders and the change in the priority level of projects; (5) the length of multi-project period will change with customer demands. In order to guarantee that the order with higher priority is delivered on time, the ones with lower priority has to be postponed to release the occupied resource; (6) penalty and rewards exist if the project is postponed or finished in advance respectively, the amounts also should be related to the priority level of projects; (7) the allowed time is limited; (8) working overtime is allowed but should be limited within the scope of labor laws. In this paper, the overtime each day should be less than 2 hours. By the way, overtime wage is 1.5 times higher than the cost of normal working hours. According to the above mentioned features of multiple R&D projects in large make-to-order enterprises and the resource-constrained multi-project model proposed by Wiest and Levy [1], and combined with the improvement made and methods raised by Wiley et al. [2], planned multiple R&D project scheduling model which meets the make-to-order environment is built. However, Wiest and levy only consider the crashing cost and rewards for finishing the project in advance, while the postpone penalty caused by the occupation of other projects has not been considered. Wiley improves the model and also considers the crashing cost, but he also uses the dual code as Wiest and Levy, that is, trimming activity ($i$, $j$) as a project activity between node $i$ and node $j$, which is difficult to understand and increases the complexity of the model.

Accordingly, RCPSP has been treated by many approaches. Particle swarm optimization (PSO) is the most common one, which is proposed by Kennedy and Eberhart [3]. PSO is a population based optimization algorithm, and has exhibited many successful applications, ranging from evolving weights and structure for artificial networks [4–6], manufacturing and milling [7], and reactive power and voltage control [8], and estimation to electric power distribution systems [9]. Since particle swarm algorithm is easy to trap into partial optimization and it is difficult to find a solution for multiple problems, many experts have made a lot of improvements to it. Zhang et al. [10] analyze the constrained project scheduling problem by using the original particle swarm algorithm, and then achieve two different scheduling generating methods. And research on the application of particle swarm algorithm to constrained multi-project scheduling problem is attracting more attention than ever. Sha and Hsu [11] apply Tabu search to improve the solution quality. Yin et al. [12] embed a hill-climbing heuristic in the iterations of the PSO. Jiao et al. [13] use the dynamic inertia weight that decreases along with iterations. Liu et al. [14] propose a model in which the center particle is incorporated into the linearly decreasing weight particle swarm optimization (LDWPSO). Unlike other ordinary particles in LDWPSO, the center particle has no explicit velocity and is set to be the center of the swarm at every iteration. Other aspects of the center particle are the same as the ordinary particle. Valls et al. [15] propose a hybrid genetic algorithm (HGA) for the resource-constrained project scheduling problem (RCPSP) with the implication of a local improvement operator and a new way to select the parents. Similarly, based on random keys, Gonçalves et al. [16] present a genetic algorithm for the resource-constrained multi-project scheduling problem, and schedules are conducted by using a heuristic algorithm that builds parameterized active schedules contingent on priorities, delay times, and release dates. Elloumi and Fortemps [17] transform the problem of single objective MRCPSP to a biobjective one to cope with the potential violation of nonrenewable resource constraints and build the fitness function as an adaptive one relying on clustering techniques, aiming to analyze more relevant fitness values. Coelho and Vanhoucke [18] provide a new algorithm, which splits the problem types into mode assignment and single mode project scheduling: the former is solved by a satisfiability (SAT) problem solver and returns a feasible mode selection to the project scheduling step; and the latter is solved by using an efficient metaheuristic procedure to work out the resource-constrained project scheduling problem (RCPSP). Then they execute these two steps in one run with a single priority list, which is different from many traditional metaheuristic methods. Afshar-Nadjafi et al. [19] develop a metaheuristic algorithm, namely, the genetic algorithm (GA), to obtain a global optimum solution or at least a partial one, and then employ the Taguchi experimental design as a statistical optimization technique to calibrate the effective parameters. Furthermore, in the other areas, by the aid of the particle swarm optimization (PSO), optimizing a GA-SVM method [20], predicting single nucleotide polymorphisms (SNPs) and selecting tag SNPs [21] or solving the heating system planning problem [22] are presented. Synthetically, Zhang et al. combine PSO with SVM for classifying magnetic resonance imaging (MRI) brain images [23].

Above all, the improvement of original particle swarm optimization algorithm can be roughly grouped into four categories as follows: (1) the improvement depends on incorporating new coefficients into velocity and position equations of the PSO algorithm or rational selection to the value of coefficients; (2) a significant point of PSO algorithms is to improve the degree of information sharing among the neighbourhoods; (3) the operators of other evolutionary algorithms are combined with PSO's; (4) some mechanisms are designed to increase the diversity of particles in order to prevent premature convergence to local minimum. But the research refers to the application of particle swarm algorithm to resource-constrained multiproject scheduling problem is few. So, in this paper, we perform the part not researched by others, and wield the particle swarm algorithm with some improvements based on the fruit of predecessors.

The rest parts of the paper are organized as follows. In Section 2, we build a resource-constrained scheduling model with multiple projects in large make-to-order enterprises. The goal of minimizing R&D cost should be fulfilled under the condition that the allowed total time of R&D time cannot exceed the limited total time. Next, in Section 3, we put forward the improved dynamic center particle swarm optimization algorithm based on the original particle optimization algorithm. Then the improved algorithm, which includes the equation of the particle and calculation of adaptive values, are presented in Section 4. In Section 5, we change the dual code to single code and make additional adjustments under the given conditions and limitations for meeting the requirements of the model. Then the effectiveness of the model and the algorithm will be proved in the simulation experiment with MATLAB program. And Section 6 is the conclusion.

## 2. Model Formulation

Absorbing the advantages of above literature, as well as considering the problem under more comprehensive conditions, this paper adopts the single code to describe the network of projects and builds a multi-project R&D scheduling model to reschedule projects activities with the minimum cost under constrained resource. The definition of the variables and parameters is as follows.

Assume that the number of projects which need resource distribution is $N$, expressed as $P_i, i = 1, 2, \dots, N$. The number of activities each project involves is $m_i$. The project activities set is $P_{ij} = (P_{i1}, P_{i2}, \dots, P_{im_i})$, $j = 1, 2, \dots, m_i$. $P_{ij}$ means activity $j$ of project $i$. The start time of $P_{ij}$ is $ST_{ij}$, the finishing time is $FT_{ij}$, and the immediate predecessor activity set of $P_{ij}$ is $PS_j$. $A_t$ is the going activity set at time $t$. $T_i$ is the normal working time of project $i$. $T_{ij}$ is the normal working time of $P_{ij}$. The crash time of $P_{ij}$ owing to resource limitations will be expressed as $y_{ij}$, but it cannot exceed the top level, which is knew as $M_{ij}$. At the same time, the reward of unit time obtained for finishing the project in advance is $L_i$, and the penalty for postponing is $K_i$. And $T$ is the allowed total time of multiple projects, $T \leq \sum_{i=1}^{N} T_i$; TCD is the target completion date. There are two kinds of distributable resource. One is the renewable human resource $R^{\rho}$ and the total limitation of it is $R_K^{\rho}$. The human resource each unit time needed at normal working time of $P_{ij}$ is $r_{ijk}^{\rho}$. When working overtime, the resource remains the same, but the normal working time decreases. Another is the total project budget $R^{v}$, which is a continuous variable and nonrenewable resource. The resource budget of each project is $R_i^{v}$ ($R^{v} \leq \sum_{i=1}^{N} R_i^{v}$). The cost of unit human resource at unit normal working time is $C$, and the human resource unit time cost because of crashing could be $K_r$, $K_r > L_i$. OH means overhead cost per unit time.

R&D multi-project resource allocation model is as follows:

$$\min \text{ COST} = \text{OH} \times \max_i \max_j \left( FT_{ij} - ST_{ij} \right) + \sum_{i=1}^{N} \sum_{j=1}^{m_i} C {r_{ijk}}^{\rho} T_{ij} + \Pi_1 + \Pi_2 + \sum_{i=1}^{N} \sum_{j=1}^{m_i} K_r {r_{ijk}}^{\rho} y_{ij}. \tag{1}$$

(1) Thereinto,

$$\begin{gathered} \Pi_1 = \sum_{i=1}^{N} \sum_{j=1}^{m_i} \left[ L_i \times \left( T_i - \max_j FT_{ij} \right)^{+} \right], \\ i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, m_i, \\ \Pi_2 = \sum_{i=1}^{N} \sum_{j=1}^{m_i} \left[ K_i \times \left( T_i - \max_j FT_{ij} \right)^{-} \right], \\ i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, m_i, \end{gathered} \tag{2}$$

$$\text{Subject to : COST} \leq R^{v}, \tag{3}$$

$$\begin{aligned} &\text{OH} \times \max_j \left( FT_{ij} - ST_{ij} \right) + \sum_{j=1}^{m_i} C r_{ijk}^{\rho} T_{ij} \\ &\quad + \sum_{j=1}^{m_i} \left[ L_i \times \left( T_i - \max_j FT_{ij} \right)^{+} \right] \\ &\quad + \sum_{j=1}^{m_i} \left[ K_i \times \left( T_i - \max_j FT_{ij} \right)^{-} \right] + \sum_{j=1}^{m_i} K_r {r_{ijk}}^{\rho} y_{ij} \leq R_i^{v}, \end{aligned} \tag{4}$$

$$\max_i \max_j FT_{ij} \leq \text{TCD}, \tag{5}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{m_i} {r_{ijk}}^{\rho} T_{ij} \leq {R^{\rho}}_{k}, \tag{6}$$

$$y_{ij} \leq M_{ij}, \quad i = 1, 2, \ldots, N; \; j = 1, 2, \ldots, m_i. \tag{7}$$

In the above model, (1) is the target function, meaning the minimum cost of multiple R&D projects. Equation (2) is the rewards for finishing the project in advance and penalty for postponing the project. Restriction (3) means that the total cost of multiple projects must be less than the total project budget. Restriction (4) means that single project is also restricted by its budget. Restriction (5) means that the latest finishing time should be in front of the target completion time. Restriction (6) means the total constraints of human resource in multiple projects. Equation (7) means activities of each project are permitted finished by working overtime, but the crash time must be limited.

## 3. Model Solutions Based on Modified PSO

*3.1. Particle Swarm Optimization.* Each particle is individual and the swarm is composed of particles. The solution space of the problem discussed here is formulated as a search space. Each position in the search space is a correlated solution of the problem. Particles cooperate to find out the best position (the best solution) in the search space (the solution space).

Particle moves toward the best positions $p$ and $g$ through each iteration, which are found by particles and the swarm respectively. The particles move according to their own velocities, which are randomly generated when approaching $p$ and $g$. For each particle $i$ and dimension $j$, the velocities and positions of particles can be updated by the following equations:

$$\begin{aligned} v_i^{t+1} &= w \times v_i^{t} + c_1 \times \text{rand}_1 () \times \left( pbest_i - x_i^{t} \right) \\ &\quad + c_2 \times \text{rand}_2 () \times \left( gbest^{t} - x_i^{t} \right), \end{aligned} \tag{8}$$

$$x_i^{t+1} = x_i^{t} + v_i^{t+1}. \tag{9}$$

In (8) and (9), $v_i^{t+1}$ is the velocity of particle $i$ on iteration $t+1$ and $x_i^{t}$ is the position of particle $i$ on iteration $t$. $pbest_i$ is the best position $p$ of particle $i$ and $gbest^{t}$ is the best position $g$ of the swarm on iteration $t$, respectively. The inertia weight $w$ was first proposed by Shi and Eberhart [24] and is used to control exploration and exploitation. The particles maintain high velocities with a larger inertia weight factor $w$, and the ones with low velocities have a smaller $w$. A larger $w$ can prevent particles and the swarm from being local optima, and a smaller $w$ encourages particles to exploit the same search space area. The constants $c_1$ and $c_2$ are used to decide whether particles prefer moving toward a best position $p$ or $g$ severally. And $\text{rand}_1()$ and $\text{rand}_2()$are random vectors between 0 and 1.

Improvement has been made in light of the defect that GPSO is easily trapped into partial optimization, and a mixed particle swarm optimization algorithm has been proposed. Also the algorithm has been applied to the model of multi-project resource distribution in large make-to-order enterprises.

*3.2. Dynamic Center Particle Swarm Optimization.* After studying the methods that improve original particle swarm optimization algorithm and analyzing the experiment results,

```
Begin PSO
       Initialize ( );
        For $t = 1$ to max iteration
$Fitness_i^t = euvaluationfitness(X_i^t)$;
              if $Fitness_i^t \geq betterFitness$
$betterFitness = Fitness_i^t$
                     end
               If necessary, update $P_i$ and $P_g$
$w = rand(0.4, 0.9)$
             $c_1 = 4 \times (Maxiteration - iterations)/Maxiteration$
              $c_2 = 4 \times iterations/Maxiteration$
       for $i = 1 : N - 1$
           Update Velocity $V_i^{t+1}$ according to formula (8)
           Limit Velocity $V_i^{t+1}$ between min velocity and max velocity
           Update Position $X_i^{t+1}$ according to formula (9)
           If need be, update $P_i$ and $P_g$
           End
           Update the position of center particle according to formula (12);
           Terminate if $t =$ max $iteration$
End
The process of Pso ends.
```

Algorithm 1

a new dynamic center particle swarm optimization algorithm has been proposed. The inertia weight $w$ and learning factors $c_1$, $c_2$ in the particle velocity equation vary with iterations. The variable range of $w$ is [0.4, 0.9]. $c_1$, $c_2$ will show linear variation when the iterations increases. At the beginning, $c_1$ is high, and particle mainly refers to social learning. With the increase of iterations, $c_2$ goes up and $c_1$ goes down, and particles come down to cognitive learning. Seeking in partial scope, the values of $c_1$, $c_2$ are

$$c_1 = 4 \times \frac{(\text{Max iteration} - \text{iterations})}{\text{Maxiteration}}, \quad (10)$$

$$c_2 = 4 \times \frac{\text{iterations}}{\text{Maxiteration}}. \quad (11)$$

Meanwhile, a center particle is proposed to visit the center of swarm at each iteration explicitly. After $(N - 1)$ particles update their positions under the dynamic PSO algorithm, and their inertia weight and learning factors are varied with every iteration. Then center particle at each iteration is updated according to the following formula:

$$X_c^{t+1} = \frac{1}{N-1} \sum_{i=1}^{N-1} X_i^{t+1}. \quad (12)$$

The pseudocode of dynamic PSO is as shown in Algorithm 1.

## 4. Particle Representation and Fitness Evaluation

Generally, It has two methods to represent particles: priority of activities and sequence permutation representation. In this paper, a particle represents a set of activity priorities because it can avoid useless scheduling plan. Actually, in multiple activities, the arrangement is based on the minimum unit activity and assume the immediate predecessor set only contingent on the immediate predecessor relationship expressed by multiple networks, which is the same as that in single scheduling arrangement.

Thus we formulate a particle by the following vector:

$$\begin{gathered} P = (p_1, p_2, \dots, p_N), \\ \text{subject to: } 0 \leq p_i \leq 1. \end{gathered} \quad (13)$$

The above particle formulation (13) represented the priority of multiple projects' activities that assure minimum cost. The dynamic center PSO initializes a swarm of particles at random, and then these particles should be moved according to formulation (8)–(12) iteratively. However, the movements of the particles are constrained by adaptive resource bounds.

When particles are updating velocities and positions during iteration, new dynamic center particle swarm algorithm proposed in this paper will be adopted, and the inertia weight and two learning factors are varying too. Meanwhile, the position and velocity of the center particle remain the same.

The swarm intelligence of dynamic center PSO is determined by $p$ and $g$. Thus, "fitness" of the particle should be evaluated. We define the fitness function of particle $P$ by

$$\text{fitness}(P) = \frac{1}{J(P)}, \quad (14)$$

where $J(P)$ is computed by the objection function (1), using $P$ and scheduled with $P$. Thus, the smaller the total cost incurred by $P$, the higher the fitness, and vice versa. The dynamic center PSO enables the particle evolution toward the target with the minimum cost.

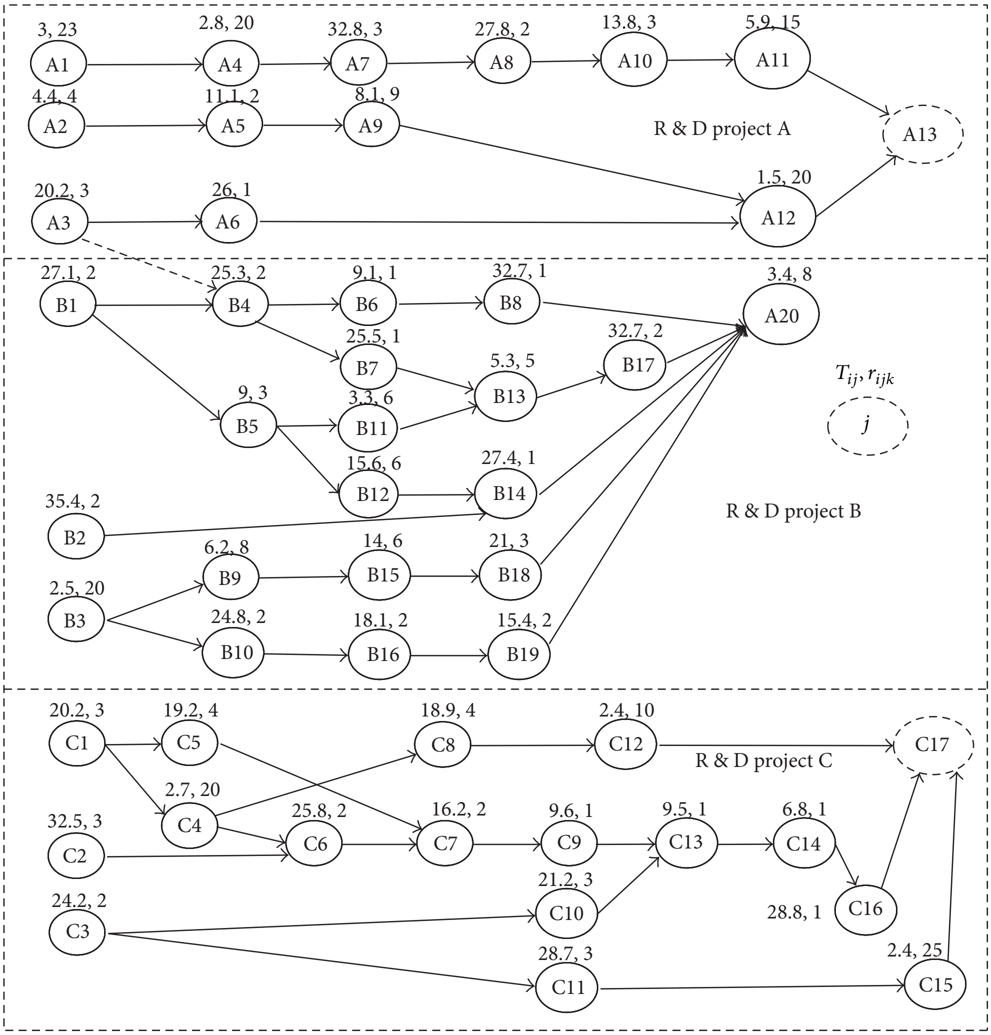


Figure 1: Code network.

In order to evaluate the adaptive values, no matter which particle it is, the elements in the particle must be transferred to a multi-project schedule. There are two ways to generate the schedule: serial scheduling scheme and parallel scheduling scheme [25, 26]. Serial scheduling scheme is arranged based on the task, while parallel scheduling scheme is arranged based on time. Generally, parallel activities of a single project are limited. So to get the optimization of scheduling, it is better to use serial scheduling scheme. But under the multi-project circumstance, there are a lot of parallel activities, so it is suggested to use parallel scheduling scheme to save time. Different from single project parallel generation, in a multi-project environment, priority should be considered in qualified activity set, which means that activities with high priority should be first selected, and then the short term ones will be chosen from them to make the schedule.

To guarantee all the orders placed to large make-to-order enterprises can be delivered in time, each R&D project must be finished before the target completion date. In case of resource constraining the project to be unfinished within normal working time, crashing is needed to shorten the normal working time. In this paper, assume that all staffs have to work overtime when necessary, and then the overworking time can be used to offset normal working time by equal number, which is eight hours per day, namely, one hour overtime means 1/8 of normal working time can be reduced. But the unit cost of crashing is 1.5 times higher than that of normal working time, that is, $K_r = 1.5C$.

## 5. Numerical Analysis

To illustrate the performance of the algorithm we proposed, several examples with a variety of datum have been investigated. In this section, dual code network has been changed to single code network as shown in Figure 1. Proper adjustment has been made in accordance with related datum. For example, a dummy has been added to the ends of two parallel activities, which is more suitable to the model. The finishing time of single project is equal to the completion time of the latest one.

Besides, the direct cost has been changed to unit time resource, and $C = 80$, $Kr = 120$, $R_1^v = 11800000$, $R_2^v = 16350000$, $R_3^v = 1370000$, $\text{OH} = 305000$, $R_k^\rho = 25$.

In the experiment, the maximum iterations of dynamic center particle optimization algorithm is 500, the scale of species group is N = 40, $w = \text{rand}(0.4, 0.9)$, $c_1 + c_2 = 4$ and varies with iterations in accordance with formulation (10) and (11). After repeated experiments, the results can be obtained: the minimum working time is 78 months, and the minimum cost is RMB 57662330.

## 6. Conclusion

This paper discusses the features of multi-project R&D environments in large make-to-order enterprises with the demand of resource distribution, and makes some improvements to algorithms and models researched before, then puts forward a multi-project scheduling model with crashing cost and postponing punishment. At the same time, based on the analysis of the particle swarm optimization algorithm and its improvement, a new solution to multiple projects scheduling has been arisen. The best solution can be obtained after the experiment to the illustration example through MATLAB program, which proves the correctness of the model we built and the improvements we made in the algorithm.

However, we assume that it has a lot to do to make the study further and deeper, because in this paper, when doing the simulation calculation, the total working time of multiprojects is assumed to be fixed in the first place, and the crashing time will be considered with the methods of permutation and combination. Moreover, the optimal scheme can be achieved only when that all the projects can be finished under constrained resource. Additionally, the uncertainty of demands to resource has not been considered because of the limitation of space and time. All these are worth to be explored in the future.

## Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors would like to thank all the editors and referees for their valuable comments. This work was supported by the National Natural Science Foundation of China (nos. 71102174 and 71372019), Specialized Research Fund for Doctoral Program of Higher Education of China (no. 20111101120019), Beijing Philosophy and Social Science Foundation of China (no. 11JGC106), Beijing Higher Education Young Elite Teacher Project (no. YETP1173), and China Postdoctoral Science Foundation (no. 2013M542066).

## References

[1] J. D. Wiest and F. K. Levy, *Management Guide to PERT/CPM*, 1969.

[2] V. D. Wiley, R. F. Deckro, and J. A. Jackson Jr., "Optimization analysis for design and planning of multi-project programs," *European Journal of Operational Research*, vol. 107, no. 2, pp. 492–506, 1998.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, IEEE Service Center, Piscataway, NJ, USA, December 1995.

[4] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE World Congress on Computational Intelligence, IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.

[5] R. C. Eberhart and Y. Shi, "Evolving artificial neural networks," in *Proceedings of the International Conference on Neural Networks and Brain*, pp. 84–89, 1998.

[6] P. H. H. Qinglan, "A BP neural networks learning algorithm research based on particle swarm optimizer," *Computer Engineering and Applications*, vol. 16, article 013, 2006.

[7] V. Tandon, *Closing the gap between CAD/CAM and optimized CNC end milling [M.S. thesis]*, Purdue School of Engineering and Technology, Indiana University Purdue University Indianapolis, 2000.

[8] H. Yoshida, K. Kawata, Y. Fukuyama, S. Takayama, and Y. Nakanishi, "A Particle swarm optimization for reactive power and voltage control considering voltage security assessment," *IEEE Transactions on Power Systems*, vol. 15, no. 4, pp. 1232–1239, 2000.

[9] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "A hybrid particle swarm optimization for distribution state estimation," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 60–68, 2003.

[10] H. Zhang, H. Li, and C. M. Tam, "Particle swarm optimization for resource-constrained project scheduling," *International Journal of Project Management*, vol. 24, no. 1, pp. 83–92, 2006.

[11] D. Y. Sha and C.-Y. Hsu, "A hybrid particle swarm optimization for job shop scheduling problem," *Computers & Industrial Engineering*, vol. 51, no. 4, pp. 791–808, 2006.

[12] P.-Y. Yin, S.-S. Yu, P.-P. Wang, and Y.-T. Wang, "A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems," *Computer Standards & Interfaces*, vol. 28, no. 4, pp. 441–450, 2006.

[13] B. Jiao, Z. Lian, and X. Gu, "A dynamic inertia weight particle swarm optimization algorithm," *Chaos, Solitons & Fractals*, vol. 37, no. 3, pp. 698–705, 2008.

[14] Y. Liu, Z. Qin, Z. Shi, and J. Lu, "Center particle swarm optimization," *Neurocomputing*, vol. 70, no. 4–6, pp. 672–679, 2007.

[15] V. Valls, F. Ballestín, and S. Quintanilla, "A hybrid genetic algorithm for the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 185, no. 2, pp. 495–508, 2008.

[16] J. F. Gonçalves, J. J. M. Mendes, and M. G. C. Resende, "A genetic algorithm for the resource constrained multi-project scheduling problem," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1171–1190, 2008.

[17] S. Elloumi and P. Fortemps, "A hybrid rank-based evolutionary algorithm applied to multi-mode resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 205, no. 1, pp. 31–41, 2010.

[18] J. Coelho and M. Vanhoucke, "Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers," *European Journal of Operational Research*, vol. 213, no. 1, pp. 73–82, 2011.

[19] B. Afshar-Nadjafi, A. Rahimi, and H. Karimi, "A genetic algorithm for mode identity and the resource constrained project scheduling problem," *Scientia Iranica*, vol. 20, no. 3, pp. 824–831, 2013.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

[21] T. Ilhan and G. Tezel, "A genetic algorithm-support vector machine method with parameter optimization for selecting the tag SNPs," *Journal of Biomedical Informatics*, vol. 46, pp. 328–340, 2013.

[22] R. J. Ma, N. Y. Yu, and J. Y. Hu, "Application of particle swarm optimization algorithm in the heating system planning problem," *The Scientific World Journal*, vol. 2013, Article ID 718345, 11 pages, 2013.

[23] Y. Zhang, S. Wang, G. Ji, and Z. Dong, "An MR brain images classifier system via particle swarm optimization and kernel support vector machine," *The Scientific World Journal*, vol. 2013, Article ID 130134, 9 pages, 2013.

[24] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, pp. 591–600, Springer, Berlin, Germany, 1998.

[25] R. Kolisch, "Serial and parallel resource-constrained project scheduling methods revisited: theory and computation," *European Journal of Operational Research*, vol. 90, no. 2, pp. 320–333, 1996.

[26] G. Li, F. Lv, and X. Guan, "Collaborative scheduling model for the Supply-Hub with multiple suppliers and multiple manufacturers," *The Scientific World Journal*, vol. 2014, Article ID 894573, 12 pages, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 164031, 15 pages
http://dx.doi.org/10.1155/2014/164031

*Research Article*

# Optimization and Planning of Emergency Evacuation Routes Considering Traffic Control

**Guo Li,[1,2] Lijun Zhang,[1,2] and Zhaohua Wang[1,2]**

[1] *School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China*
[2] *Center for Energy and Environmental Policy Research, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Guo Li; lg4229682@163.com

Received 25 October 2013; Accepted 24 December 2013; Published 29 May 2014

Academic Editors: T. Chen and J. Yang

Emergencies, especially major ones, happen fast, randomly, as well as unpredictably, and generally will bring great harm to people's life and the economy. Therefore, governments and lots of professionals devote themselves to taking effective measures and providing optimal evacuation plans. This paper establishes two different emergency evacuation models on the basis of the maximum flow model (MFM) and the minimum-cost maximum flow model (MC-MFM), and proposes corresponding algorithms for the evacuation from one source node to one designated destination (one-to-one evacuation). Ulteriorly, we extend our evaluation model from one source node to many designated destinations (one-to-many evacuation). At last, we make case analysis of evacuation optimization and planning in Beijing, and obtain the desired evacuation routes and effective traffic control measures from the perspective of sufficiency and practicability. Both analytical and numerical results support that our models are feasible and practical.

## 1. Introduction

Major emergencies, concerning accidents such as natural disasters, public health events, and social abrupt affairs, occur in a very short period of time and always cause serious damage to the society. They are all characterized by suddenness, uncertainty, and serious harmfulness, and have been a big challenge for sustainable development of the human society.

In recent years, various disasters occur frequently. Therefore, countries all over the world attach great importance to emergency management. China has entered into the high-incidence season of emergencies and will face the baptism brought by emergencies in a long period of time [1]. With the further acceleration of urbanization, the population is more concentrated, so how to respond effectively and timely to emergencies is especially important. In most emergency management, how to evacuate people to safety zone becomes a crucial step. For example, cyclohexane spill that occurred in 1976 in Seveso, Italy, rained a cloud of dioxin on surrounding communities, and the local authority organized a evacuation of 220,000 people. Similarly, when liquid chlorine cylinder explosion accident occurred in southern China in 1979, emergency evacuation was also implemented to move 60,000 people away. Likewise, in 1984, after Mexico City gas storage tank exploded, 350,000 people were evacuated [2].

The evacuation issues can be divided into two different types: small scale evacuation and long-distance regional evacuation [3]. Small scale evacuation generally refers to a kind of evacuation when emergencies affect small area swiftly and violently, such as explosion within a finite range, house collapsing, fire breaking out in shopping malls, and so on. Generally, evacuation of this kind mainly deals with evacuation on foot, rather than using vehicles. In contrast, long-distance regional evacuation refers to a form of evacuation implemented along with the appearance of a wide range of emergencies, such as leakage and diffusion of harmful gases and earthquakes. Since the evacuation route is long-distance, this kind of evacuation generally needs vehicles for transport. In addition, according to the difference of subject and actions taken after emergencies happen, the long-distance regional evacuation can be divided into autonomous evacuation, recommended evacuation, and mandatory evacuation [4]. Autonomous evacuation and recommended evacuation are carried out spontaneously and preparedly by people when they get messages or notifications of disasters, in which case the alert time is comparatively longer, while mandatory evacuation responds to emergencies having the need of urgent evacuation and generally commanded by governments or related departments. All of the evacuations considered in this paper belong to the latter situation.

The rest of our paper is organized as follows. Section 2 presents a review of related literature. Section 3 deals with one-to-one evacuation optimization problem based on MFM. Section 4 develops one-to-one evacuation model based on MC-MFM. Section 5 explores the one-to-many evacuation problem. Case analysis is reported in Section 6, and finally some concluding remarks are given in Section 7.

## 2. Literature Review

Generally, scholars mainly tend to consider long-distance regional evacuation or mandatory evacuation first, figuring evacuation routes as network diagrams to achieve shortest routes, maximum flow, or minimum-cost maximum flow, such as adopting contraflow transportation network reconfiguration [5], establishing the cell-based evaluation network model [6], analyzing multifactors with the construction of a system which involves the evacuation network [7], building the network flow model for lane-based evacuation routing [8], and solving evacuation problems by using dynamic network flows [9, 10]. Lots of researchers have not only optimized evacuation route, but also explored many methods. The representatives refer to Choi et al. [11], Han [12], Hoppe and Tardos [13], Klüpfel et al. [14], Theodoulou [15], and Adewmi and Garba [16]. With regard to multiobjective evacuation routing, Stepanov and Smith [17] present a methodology for designing optimal routing policies for emergency evacuation planning (EEP) through building an integer program (IP) model. On the problem of roadway deployment, Pal and Bose [18] propose an integer model to find the best location and assign response vehicles to those depots under reliability constraints. Saeed and Ram [19] examine the evacuation routing problems by proposing a two-step approach that consists of an incapacitated integer multi-commodity network model and a computational algorithm. Considering nonlinear relations, Zhang et al. [20] explore a min-cost-pursued swapping dynamic (NMSD) system to model the evolution of selfish routing games on the traffic network.

Besides, many topics related to emergency evacuation also have been discussed. Regarding the evacuation problem in surrounding areas of nuclear power plant, Dunn and Newton [21] investigate how to evacuate people as far as possible within the prescribed time, looking upon the evacuation problem as a maximum flow problem to find the optimal evacuation route. As to the evacuation in an earthquake, Yamada [22] describes route optimization of neighborhoods as the shortest route problem and the minimum cost flow problem, respectively. Campos and da Silva [23] try to reduce the conflicts between finding the evacuation flows and increasing the traffic capacity of evacuation route. They treat the ratio of traffic capacity and travel time as a measurable index of traffic performance. On the issue of hurricane evacuation, Dixit and Radwan [24] provide a new and innovative technique with a "network breathing strategy" at destinations after dictating when to schedule evacuation orders and capacities required on different routes. Based on the data of earthquakes in 1994, Northridge and 1995, Kobe, Koike et al. [25] provide a probabilistic approach to simulate the evacuation scenario along the streets crowded with evacuation people. Specifically, the ratio of fire-proof structures, ignition rates from fire gutted houses, and population density are all assumed in a probabilistic manner. In view of blocking effects on crowd movement, Luh et al. [26] establish a new macroscopic network-flow model assuming that fire, smoke, and psychological factors can evoke crowd's desire to escape at the expected flow rate and then develop a divide-and-conquer approach to reduce computational complexity and reflect psychological changes.

As for effective solutions for route optimization, Hamacher and Tjandra [27] all systematically summarize the dynamic network flow models, which are widely used in the optimization of human organization evacuation planning, including the dynamic maximum flow model, the fastest dynamic flow model, and the global maximum dynamic flow model. Besides, they analyze the utilization potentiality of these models in regional and extensive evacuation. Mamada et al. (2003) [28] treat the optimization of evacuation routes and the assignment of departure times as the fastest flow problem. With the assumption that the traffic starting from the source node could only depart for a designated destination, they provide a algorithm to find the fastest flow in the tree network diagram. Similarly, Lu et al. [29] regard the evacuation planning that includes departure time, evacuation routes, and ending point selection as the fastest transshipment problem, and propose a heuristic algorithm on the basis of network figural representation to find the optimal organization planning directly. Comprehensively, Lin [30] constructs a framework integrated with optimizations, evaluations, and adjustments, aiming to achieve the optimization of evacuation organization planning. Fuellerer et al. [31] consider that, on two-dimensional level, successive routing of the vehicles in the road network by a colony algorithm can satisfy customers' demands with the consideration of the freight loading factor. Likewise, Leung et al. [32] develop an extended guided tabu search (EGTS) method and a new heuristic packing algorithm for the two-dimensional loading vehicle routing problem, which can help tabu search to escape from local optimum effectively. From the perspective of information quality and evolution, Pillac et al. [33] present a general description of dynamic routing and introduce the notion of degree of dynamism, and then bring out a comprehensive review of applications and solution methods for dynamic vehicle routing problems. When capacity constraints exist, Xu et al. [34] propose a stochastic user equilibrium assignment model for a schedule-based transit network, through which they can simultaneously predict how passengers choose their transit vehicles to minimize their travel costs and estimate the associated costs. And they also find that when a connection segment reaches its capacity level, the Lagrange multipliers of the mathematical problem give the equilibrium passenger overload delays in this transit network.

In addition, some scholars formulate models to find the concrete flow and path from source nodes to designated destinations. To minimize the end-to-end delays for the specific routing mechanism, Grimmell and Rao [35] discuss the problem that deals with the transmission of a message from a source node to a designated destination over a network with propagation delays and dynamic bandwidth constraints on the links. They make available bandwidth for each link specified as a piecewise constant function and present for each message forwarding. In order to identify the network paths accurately, Murtaza et al. [36] present a new mechanism for detecting shared bottlenecks between end-to-end paths in a network, which is based on the well-known linear algebraic approach-singular value decomposition (SVD). Specific to the evacuation in the feeder-bus network, Deng et al. [37] extend the demand pattern M-to-1 (i.e., multiple bus stops and a single station) to M-to-M, considering the passenger travel cost. Moreover, they present a new genetic algorithm to determine the optimal feeder-bus operating frequencies under strict constraints, and finally find that demand distributions also should be considered when designing a feeder-bus network.

The literature mentioned above mostly converts evacuation problems to dynamic or static network model by finding shortest routes, maximum flow, and minimum-cost maximum flow. However, intuitive calculation makes the entire emergency management on evacuation too rigid, and there are many actual-world factors that cannot be fully reflected in the model.

First of all, intersections are everywhere in reality, so the impacts of them should be considered during evacuation process in avoidance of possible time delays. Although it may make computing process more complex, it is better than ignoring intersections' impact. By the way, the complexity of intersections does not only reflect on the limitation of certain straight road sections, but also on that of different turning routes. Being different from classic emergency evacuation management, this paper fully examines the influence of intersections.

Secondly, previous researches on impacts of traffic management and control measures are very few. The actual evacuation may be affected by many aspects of constraints, which are all objective conditions that constrain the evacuation carried on. Therefore, control measures should be considered in the model, such as duplicated row, single row, and forbidding for passing as well as some uncertain real-time management and control measures, like signal control and real-time directing.

Finally, regarding the problem of establishing evacuation models, two different kinds of evacuation, namely, evacuations from one source node to one designated destination(one-to-one evacuation) and one source node to multiple designated destinations (one-to-many evacuation), also are taken into account according to the actual situation.

## 3. Evacuation Optimization Model Based on MFM

When dangerous situations are unknown to people, the maximum number of evacuations can only be implemented within the capacity limitations, and each batch of evacuation must be the maximum flow. In addition, in order to facilitate the description and calculation, the maximum flow below generally refers to the flow.

*3.1. Assumptions and Prerequisites.* In the network diagram model with the given time and the goal of maximum flow, its basic assumptions and prerequisites are as follows.

(1) There is only one source node and one designated destination, and vehicle number in the source node are known.

(2) Allowed evacuation time can be obtained by forecasts, and thus evacuation time is given.

(3) Traffic capacity of each road section and each intersection's different turnings are known.

(4) Limitations of traffic capacity are only valid for one batch, and next batch has new limitations on its traffic capacity.

(5) The goal is to maximize evacuation vehicle number under limitations of evacuation time and traffic capacity, namely, to find the maximum flow in a network diagram.

In the above prerequisites, assumption (4) also reflects a characteristic of this modeling. Maximum-flow model generally refers to the model for achieving one-time maximum flow under traffic capacity constraints. Due to the limitations of traffic capacity, maximum flow in multiple times is taken into account in this model. Under this circumstance, batches will be limited, and each batch has the same limitation of traffic capacity. Moreover, the maximum flow includes vehicle number of the $m$th batch and the $(m-1)$th batch, where $m = \lceil T/t \rceil$.

*3.2. Notations.* Nodes include source nodes, intersections, and evacuation destinations. Arcs represent road sections between the intersections, source nodes, destinations, and adjacent intersections, which convert the entire road network to directive network diagrams $G = (V, L, C, U)$, where $V$ represents a vertex in the figure, expressed by characters $i, j, k$. The node 1 and node $n$ represent the source node and the designated destination, respectively.

$L$ represents the arc, while $(i, j)$ represents straight line arc from $i$ to $j$, and $(i, j, k)$ represents steering arc from $(i, j)$ to $(j, k)$ through $j$.

$C$ stands for the node weight set, and $c_{ijk}$ represents the traffic capacity of $j$ from road section $(i, j)$ to road section $(j, k)$ through $j$.

$U$ is the arc weight set, and $u_{ij}$ stands for the traffic capacity of road section $(i, j)$.

$m$ represents the final evacuation batch.

$f^0$ is the given flow of source node.

$f_i$ is the maximum flow of $i$th batch.

$f_m$ is the flow of final batch evacuation.

$f_T$ represents the total number of evacuation flow.

$f_{ij}$ represents the flow on arc $(i, j)$.

$f_{ijk}$ represents the flow from the arc $(i, j)$ to arc $(j, k)$ through $j$.

*3.3. Formulations Based on MFM.* Based on the above assumptions and definitions, this evacuation can be expressed as the following models:

$$\text{Max} \quad f_T = \sum_{i=1}^{m} f_i \tag{1}$$

$$= (m-1) f_i + f_m$$

$$= (m-1) \sum_{i=1}^{n} f_{1i} + f_m \tag{2}$$

$$\text{s.t.} \quad 0 \le f_{ij} \le u_{ij}, \quad (i, j) \in L \tag{3}$$

$$0 \le f_{ijk} \le c_{ijk}, \quad (i, j), (j, k) \in L \tag{4}$$

$$\sum_{i=1}^{n} f_{1i} = \sum_{i=1}^{n} f_{in} \tag{5}$$

$$f_{ij} = \sum_{k=1}^{n} f_{ijk}, \quad (i, j) \in L \tag{6}$$

$$f_{jk} = \sum_{i=1}^{n} f_{ijk}, \quad (j, k) \in L \tag{7}$$

$$0 \le f_m \le f_i \tag{8}$$

$$f_{ij} \ge 0, \quad f_{ijk} \ge 0 \; (i, j), \quad (j, k) \in L. \tag{9}$$

While formula (1) is the objective function, formula (2) is its decomposition type; formulas (3) and (4) are constraints of traffic capacity of road sections and intersections, respectively; formula (5) means that the flow of each batch reaching the destination equals the flow departing from the source node; formulas (6) and (7) are the flow conservative constraints of each intersection; formula (8) represents that the flow of the final batch is not more than the maximum evacuation flow of front batches; formula (9) is nonnegative constraints.

*3.4. Solutions.* The model established above is not linear, so linear programming cannot be used directly to find the solution. Certainly, this model can be converted into one or multiple linear programming models, but obviously the solution process will be more complex. Comparatively speaking, it is much more convenient to establish the network diagram model.

From the view of formula 1, the key point of the solution is to obtain $m$, $f_1$, and $f_m$.

In fact, $f_1$ can be obtained by the MFM in the network diagram, using labeling method. Traditional solution to find maximum flow in network diagram is to label points in sequence to obtain the augmented chain. However, the solution of this model is to label arcs. Besides, arcs can be labeled including straight line arcs and steering arcs at the intersection on road sections. In this model, let $p$ represents the set of the augmented routes; let $P^+$and $P^-$ represent the set of forward arcs and backward arcs in the augmented route, respectively. Let $p[ij]$ or $p[ijk]$ represent the prior label of arc $ij$ or arc $ijk$ on augmented chain. Steps of the calculation are as follows.

*Step 1.* Give an initial feasible flow in the network. Zero flow can be treated as the initial feasible flow.

*Step 2.* Label the arc and find an augmented chain.

(1) Label a random arc that regards the source node as the starting point with $\{\infty, h_{ij}\}$, while $h_{ij}$ is the remaining possible maximum flow of the arc.

(2) Pick the next arc linked to the prior labeled arc and then check it. (a) if the arc is the forward arc and the flow of the arc is less than the traffic capacity or (b) if the arc is backward arc and nonzero arc, then label it with $\{p[ij], h_{ij}\}$ or $\{p[ijk], h_{ijk}\}$. $p[ij]$ or $p[ijk]$ is the prior labeled arc of the arc, and $h_{ij}$ or $h_{ijk}$ represents the remaining flow of the arc.

(3) If a labeled arc's endpoint is the destination, it represents the augmented chain has been found. When continuous labeling cannot make the labeled arc point to the destination, it indicates that the augmented chain does not exist, so the calculation ends.

*Step 3.* Adjust the flow.

(1) Find the minimum value of arc $h_{ij}$ or arc $h_{ijk}$ on the augmented chain, and denote it with $h$.

(2) Adjust the flow according to the following formulas:

$$f_{ij} = \begin{cases} f_{ij} & (i, j) \in P \\ f_{ij} + h & (i, j) \in P^+ \\ f_{ij} - h & (i, j) \in P^- \end{cases}. \tag{10}$$

or

$$f_{ijk} = \begin{cases} f_{ijk} & (i, j) \in P \\ f_{ijk} + h & (i, j) \in P^+ \\ f_{ijk} - h & (i, j) \in P^- \end{cases} \tag{11}$$

Then obtain the new possible flow $f_1$.

*Step 4.* Cancel all labels and repeat Steps 2 and 3 constantly until new augmented chain cannot be found. At this moment, sets consisting of $f$ of each arc are the maximum flow sets.

In fact, when maximum flows of each batch are implemented, we can know the time spent in each evacuation, as well as the evacuation time of each possible route. In the final evacuation, it is obvious that not all routes were able to evacuate in place, which means that evacuation routes consuming shorter time will be chosen preferentially. For example, before $m$th batch evacuation begins, it remains about time $q$ after prior $m$-1 batches, and then the last evacuation batch only aim at the route with the evacuation time less than $q$, and $f_m$ is the flow of final batch evacuation.

## 4. Evacuation Model Based on MC-MFM

Our evacuation model investigates a comparatively less clamant emergency situation, such as slight toxic gas leak. Everyone is required to be evacuated out of the danger zone. Here, people can rationally implement evacuation route planning and establish models with the goal of minimizing total evacuation time and maximizing the evacuation vehicle number. So evacuation problems in practical can be abstracted as a kind of minimum-cost maximum flow model (MC-MFM). Evacuations in multiple batches are still considered in this model, and only subsequent evacuation routes are implicit in the diagram. Specific model is as follows.

*4.1. Premise and Hypothesis.* In the network model with the object of obtaining the maximum flow within the given evacuation time, its basic premise and hypothesis are as follows.

(1) Only one source node and one designated destination exist, and the number of evacuation vehicles are known.

(2) Evacuation intervals of each batch are given and known.

(3) Traffic capacity of each road section and each intersection are known.

(4) The average traffic time on each road section and the average delay of different turnings of each intersection are known.

(5) The goal is to minimize the total evacuation time and maximize evacuation vehicle number.

*4.2. Symbol Definition.* Nodes include source node, intersections, and evacuation destination. Similarly, arcs include road sections between the intersections, evacuation source node, and destination. Therefore, the network can be converted to network diagram $G = (V, L, D, C, T, U)$.

$V$ is the vertex, expressed by characters $i, j, k$.

$L$ is the arc, while $(i, j)$ is the arc from $i$ to $j$.

$D$ is a point weight set, while $d_{ijk}^n$ represents the delay of road section $(i, j)$ to the road section $(j, k)$ through $j$ in $n$th batch.

$C$ is also a point weight set, while $c_{ijk}$ represents the traffic capacity of road section $(i, j)$ through $j$ to the road section $(j, k)$.

$T$ represents the arc weight set, while $t_{ij}^n$ represents average travel time of road section $(i, j)$ in $n$th batch.

$U$ represents the arc weight set, while $u_{ij}$ represents the traffic capacity of road section $(i, j)$.

$f^0$ is the total number of vehicles at source node.

$f_{ij}^n$ stands for the flow of arc $(i, j)$ in $n$th batch.

$f_{ijk}^n$ represents the flow of arc $(i, j)$ to arc $(j, k)$ through $j$ in $n$th batch.

$t$ is the evacuation interval in each batch, also known as the valid time of traffic.

Besides, node 1 and node $m$ stand for the source node and the designated destination, respectively.

*4.3. Model Formulation.* The above evacuation problem based on MC-MFM can be expressed as the following mathematical model:

$$\text{Min} \quad z(f) = \sum_{r=1}^{\infty} t_{ij}^r f_{ij}^r + \sum_{r=1}^{\infty} d_{ijk}^r f_{ijk}^r, \quad (i, j), (j, k) \in L \tag{12}$$

s.t.

$$0 \le f_{ij}^n \le u_{ij}^n, \quad (i, j) \in L \tag{13}$$

$$0 \le f_{ijk}^n \le u_{ijk}^n, \quad (i, j) \in L \tag{14}$$

$$\sum_{r=1}^{\infty}\sum_{i=1}^{n} f_{1i}^r = \sum_{r=1}^{\infty}\sum_{i=1}^{n} f_{im}^r = f^0 \tag{15}$$

$$f_{ij}^n = \sum_{k=1}^{m} f_{ijk}^n \quad (i, j) \in L \tag{16}$$

$$f_{jk}^n = \sum_{i=1}^{m} f_{ijk}^n \quad (j, k) \in L \tag{17}$$

$$f_{ij}^n \ge 0, \quad f_{ijk}^n \ge 0 \quad (i, j) \in L, \quad (j, k) \in L. \tag{18}$$

While formula (12) is the objective function; formulas (13) and (14) are the traffic capacity constraints of road sections and intersections, respectively, which means that the flow of road sections and intersections does not exceed the traffic capacity; formula (15) is the constraints of aggregate demand of evacuation, which illustrates that the total flow to the destination is equal to the total flow departing from source node; formula (16) refers to the flow of evacuation road section $(i, j)$ in $n$th batch evacuation is equal to the total flow turning to other road sections at node $j$; formula (17) means that the flow of road sections $(j, k)$ in the $n$th batch evacuation is equal to the total flow turning from other road sections to this road section through node $j$, and formula (18) is nonnegative constraints.

*4.4. Solutions.* Likewise, this model can also be converted to linear programming and solved by simplex method, but obviously it is easier to be solved by our MC-MFM.

If we find the first minimum cost flow $M$ with the cost of $a$, then we assume that there are numerous other cost flows existing with costs of $a + t, a + 2t$, and so on. Each time a feasible flow with minimum cost is found, we assume that there are numerous virtual routes. And hereby we declare that traffic capacities of these virtual routes are recovered and have the same value as the first batch route. In the process of looking for smaller cost flows, if a smaller cost is not less than $a + t$, then we take flow $M$ in the second batch as the minimum cost possible flow, and so on. Let $P$ represents the augmented route or the minimum cost route in the process of solution. $P^{+}$ and $P^{-}$ are the set of forward arcs and backward arcs in the augmented route. $p[i]$ is the prior node of node $i$ in the augmented route or the minimum cost route. The steps of calculation are as follows.

*Step 1.* Give an initial feasible flow $f_0$, which can be zero.

*Step 2.* Find a minimum cost road $p$ from node 1 to node $m$ in this network diagram.

*Step 3.* Augment the unit flow along route $p$ and add flow $h$ in augmented route $P^{+}$or intersection turnings. Otherwise, reduce flow $h$ in augmented route $P^{-}$ or intersection turnings. Then a new possible flow $f^{1}$ is formed.

*Step 4.* After one round of augmentation, wipe off the arcs which have been saturated and then repeat Steps 2 and 3 and continue to augment until $f^{0}$ has been fully evacuated out. Then calculate the total time.

Specific calculation methods of this model will be displayed in the case analysis.

## 5. Extension to One-to-Many Evacuation

The models mentioned above are the evacuation from one source node to one designated destination. To make the study go further and deeper, the evacuation from single source node to many designated destinations will be considered in this part.

As for the evacuation from single source node to many designated destinations, the destination is no longer a single parameter as $m$ but expressed by many parameters such as $m_1, m_2, \ldots, m_q$. We only need to change formula (15) to $\sum f_{1i}^{n} = \sum\sum f_{im}^{n}$, which represents the total flow arriving at multiple destinations is equal to the total flow of the source node.

We have two ways as follows to solve this problem.

The first one is to set a virtual evacuation destination $w$ and then assume the delay time or travel time of all turning arcs and line arcs toward to arc $(m_i, w)$ are zero, and the traffic capacity is $\infty$. In this way, evacuation from one source node to many designated destinations can be turned back to one source node to one designated destination.

The second one is to solve the model of one-to-many evacuation directly. In the process of calculation, the domain of the minimum cost route becomes broader. For example, if destinations $m_1$, $m_2$, and $m_3$ exist, then we must find minimum cost routes from the source node to $m_1, m_2$, and $m_3$, respectively and make comparisons to find the minimum cost route which will be regarded as the augmented route at this time. Though this method has differences with the first one, the essence is the same.

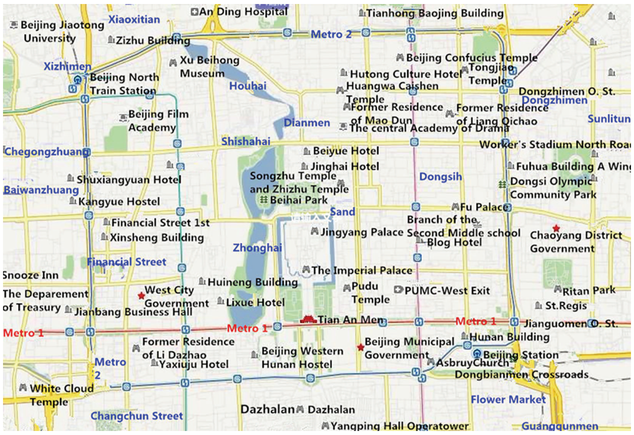


Figure 1: Beijing's second ring road.

Certainly, emergency evacuation problem in one-to-many evacuation has a little difference from one-to-one evacuation. In other words, multiple evacuation destinations may have multiple capacity limitations, which require us to consider whether the evacuation flow toward certain destination will reach the maximum capacity.

According to Figure 1, we abstract its main routes to form a network route map, see Figure 2.

## 6. Case Analysis

As we all know, Beijing is a big city with high concentration of population, especially in the downtown. Once some hazardous emergencies occur, it is likely to cause very serious losses. Under such circumstances, this paper will consider the specific route map within the second ring of Beijing, and the second ring road will be abstracted into network diagrams. Since the routes of the second ring road are comparatively complex, the abstracted route network diagrams are simplified according to importance, and more actual values for each parameter are given to ensure the accuracy of case analysis.

### 6.1. Case Description

*6.1.1. Selection of Network Diagrams.* This case analysis will calculate separately according to two different kinds of evacuation model, which is on the basis of actual second ring road map of Beijing, as shown in Figure 1.

Assume that the black spot in Figure 2 is the source node. It is located between the imperial palace and the Beihai Park, which is also the area with the most intensive stream of people. Small white circles represent the assumed possible destinations. We assume that the big circle is the range of

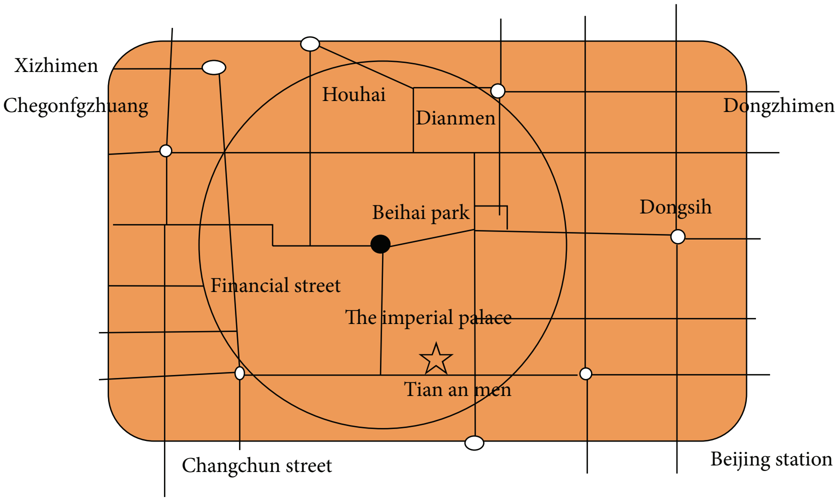


Figure 2: Route network of Beijing's second ring road.

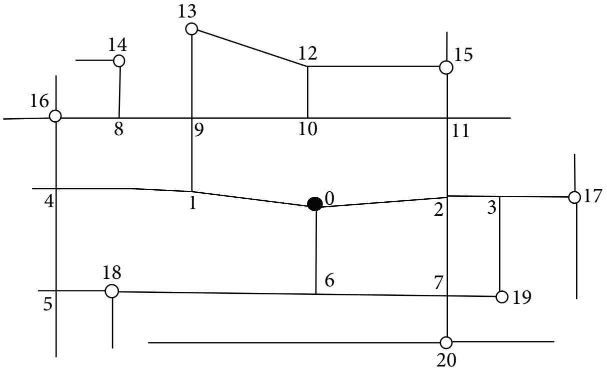


Figure 3: Simplified diagram of Beijing's second ring road traffic routes.

evacuation, and it is required that people should be evacuated to a destination or multiple destinations out of this circle.

In order to calculate more conveniently, this paper ignores some minor roads and only considers main roads, so this road map will be simplified to Figure 3.

*6.1.2. Parameter Setting.* Based on the simplified figure (Figure 3), we can conclude that there are eight T-crossroads, 1, 3, 4, 6, 8, 10, 11, and 12, and three crossroads, 2, 7, and 9.

According to the situation of actual evacuation, some road sections and intersection turnings should be regulated, and some possible traffic control measures should be implemented. Possible routes of each road section and accessible intersection turning will be described, and the traffic capacity and the amount of time consumed, including the travel time at road sections and the delay time at intersection turning, will be illustrated, see Tables 1 and 2.

Possible road sections and intersection turnings described in the above two tables are designed based on the traffic control measures. For example, there is no turning (3, 2, 11) at intersection 2 and turning (8, 9, 13) at intersection 9. In addition, we also assume that destinations will not be taken as the midpoint of other evacuation routes. For instance, there are no similar turnings as (19, 3, 17) and (15, 12, 13), and so forth. Of course, limitations of these control measures make calculations more convenient but do not affect the accuracy of the entire case.

### 6.2. Emergency Evacuation Based on MFM

*6.2.1. Problem Description.* Figure 3 is marked with multiple destinations, hereby we consider one-to-one evacuation at first and select node 13 as the destination. In order to calculate conveniently, we still assume that evacuation routes cannot pass other alternative destinations beside destination node 13. Thus, evacuation route network diagram can be simplified in Figure 4.

As the intersections 1, 6, 11, and 12 are no longer T-roads and intersections 2, 7, and 9 are also no longer crossroads, the delay time of each intersection turning will change inevitably. The traffic capacity and the amount of time consumed, including travel time and delay time of each road section and intersection turnings, are showed in Table 3.

In this paper, the total evacuation time is limited to 60 mins and the allowed evacuation time in each batch is 16 mins.

*6.2.2. Solutions.* The first thing is to obtain the value of $m$ and $m = \lceil T/t \rceil = \lceil 60/16 \rceil = 4$. The evacuation time in the fourth batch is only 12 mins.

The second thing is to get the static maximum flow $f_1$.

*Step 1.* Let initial flow $f_0$ be zero.

*Step 2.* Label arc (0, 1) with $\{\infty, 20\}$, then label (0, 1, 9) with $\{(0, 1), 16\}$; next, label (1, 9) with $\{(0, 1, 9), 16\}$ and label (1, 9, 13) with $\{(1, 9), 8\}\}$, then label (9, 13) with $\{(1, 9, 13), 16\}\}$, so the first round of labeling ends. Since $h_{1-9-13}$ is the minimum among $h_{ij}$ or $h_{jk}$, adjust it with $h = 8$, and the flow of arc (0,

Table 1: Possible road sections of emergency evacuation.

| Starting node of road sections | Possible road sections | Traffic capacity (hundreds of vehicles) | Travel time (minutes) |
|---|---|---|---|
| 0 | (0, 1) | 20 | 2 |
| | (0, 2) | 20 | 2.5 |
| | (0, 6) | 20 | 3 |
| 1 | (1, 4) | 16 | 2 |
| | (1, 9) | 16 | 2 |
| 2 | (2, 3) | 16 | 2 |
| | (2, 7) | 10 | 3 |
| | (2, 11) | 18 | 2 |
| 3 | (3, 17) | 10 | 1 |
| | (3, 19) | 10 | 3 |
| 4 | (4, 5) | 10 | 3 |
| | (4, 16) | 10 | 2 |
| 5 | (5, 18) | 8 | 2 |
| 6 | (6, 7) | 12 | 2.5 |
| | (6, 18) | 10 | 3 |
| 7 | (7, 2) | 10 | 3 |
| | (7, 19) | 16 | 2 |
| | (7, 20) | 12 | 1 |
| 8 | (8, 14) | 8 | 1 |
| | (8, 16) | 8 | 0.8 |
| 9 | (9, 8) | 12 | 1 |
| | (9, 10) | 12 | 3 |
| | (9, 13) | 16 | 3 |
| 10 | (10, 9) | 12 | 3 |
| | (10, 11) | 12 | 1.5 |
| | (10, 12) | 16 | 1 |
| 11 | (11, 10) | 12 | 1.5 |
| | (11, 15) | 16 | 1 |
| 12 | (12, 13) | 10 | 4 |
| | (12, 15) | 10 | 1.5 |

1), (0, 1, 9), (1, 9), (1, 9, 13), and (9, 13), namely, the flows on augmented chain $P$, is increased to 8. Arc (1, 9, 13) is saturated.

Label the arc with the next three rounds with adjustment.

*Step 3.* Since the limit time in the fourth batch is 12 mins. Therefore some routes, in which the travel time and the delay time are significantly more than 12 mins, will not be considered. Only route (0-1-9-13) is possible, and its flow is 8.

*Step 4.* Calculate $f_T = f_1 + f_2 + f_3 + f_4 = 26$. Specific evacuation routes are as shown in Table 4.

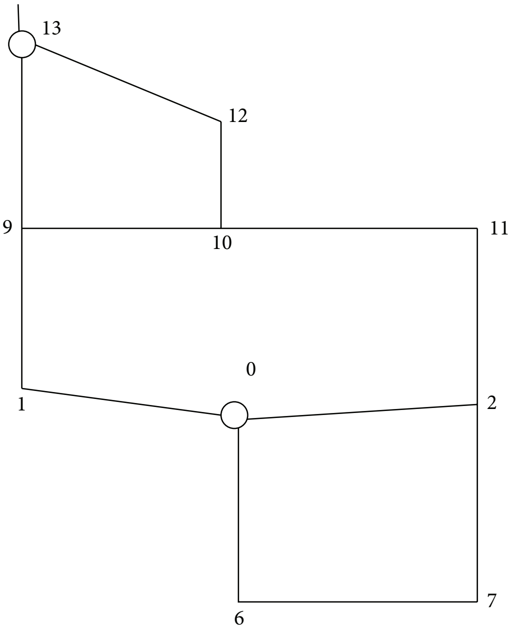


Figure 4: Simplified road sections in one-to-one evacuation.

Due to the evacuation only towards a fixed destination, the delay time of some intersections is shorter than one-to-many evacuations. In our case, the traffic control measures have been considered. For example, road section (9, 10) is a two-way traffic, and most road sections are only one-way traffic.

*6.3. Emergency Evacuation Based on MC-MFM.* Then we continue to consider one-to-one evacuation with node 13 as the destination. For the simplicity of calculation, we still assume that the evacuation route cannot pass other alternate evacuation destinations except node 13. Thus, network diagram of evacuation routes is also illustrated in Figure 4. In addition, the total number of vehicles in the source node $f_0$ is 60 hundred vehicles.

We need to find the minimum cost route and the augment route on this way until evacuation flow reaches the target value. The approach used here is almost like the Dijkstra algorithm, but the review for the routes with minimum cost has been changed into the review for arcs.

The first thing is to obtain the minimum cost route, see Table 5.

It can be seen from Table 5 that the minimum cost of route 0-1-9-13 is 10. As the bottleneck traffic flow on this route is arc (1, 9, 13), which is 8, this route is the augmented chain. When we increase its traffic flow about 8, arc (1, 9, 13) of the first batch is saturated at this moment, and the total route cost is 80, namely 8000 mins. In addition, subsequent batches of this route should be considered in the course of finding the minimum cost route.

As illustrated in Table 5, since the first batch route 0-2-11-10-12-13 does not include arc (1, 9, 13), the total cost of this route is 16, which is more than the virtual route's minimum value in the preamble, and thus the minimum cost route is route 0-1-9-13 in the second batch. As the bottleneck of traffic flow is arc (1, 9, 13), which is 8, the traffic capacity on this augmented chain is increased with 8, and arc (1, 9, 13) in the

Table 2: Possible turnings of emergency evacuation.

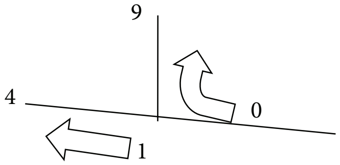
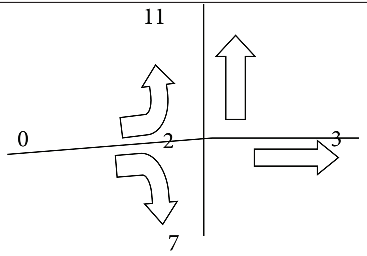
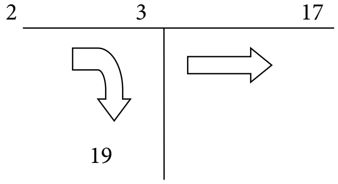
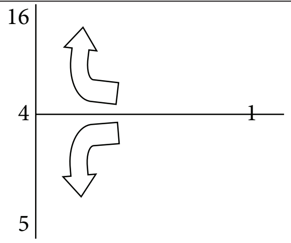
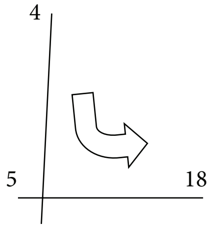
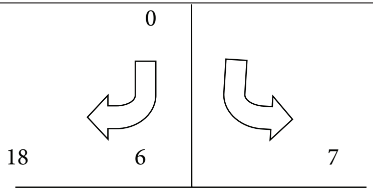
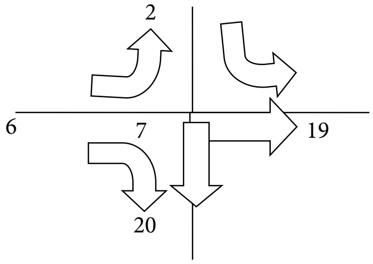
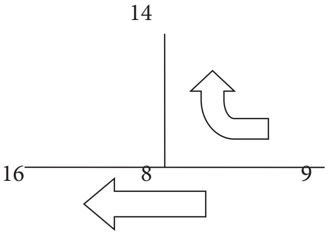
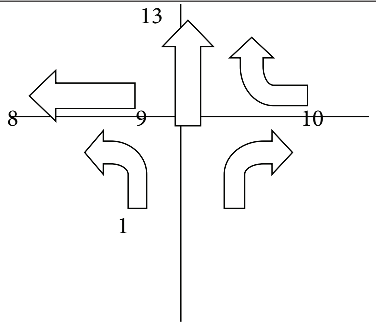
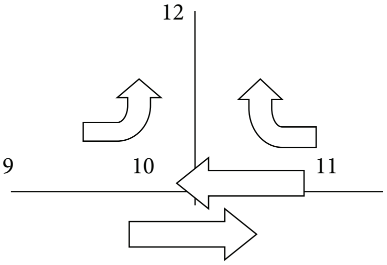
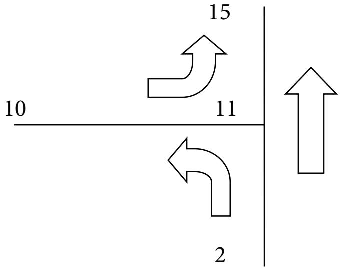
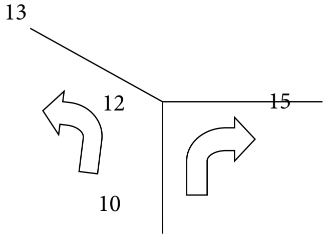
| Intersections | The graphical representation of intersections and possible turnings | Possible turnings | Traffic capacity (hundreds of vehicles) | Intersection delay (min) |
|---|---|---|---|---|
| 1 |  | (0, 1, 4)<br>(0, 1, 9) | 16<br>16 | 2<br>2 |
| 2 |  | (0, 2, 3)<br>(0, 2, 7)<br>(0, 2, 11)<br>(7, 2, 11) | 16<br>12<br>12<br>10 | 3<br>3<br>3<br>3 |
| 3 |  | (2, 3, 17)<br>(2, 3, 19) | 12<br>12 | 2<br>2 |
| 4 |  | (1, 4, 5)<br>(1, 4, 16) | 12<br>12 | 2<br>2 |
| 5 |  | (4, 5, 18) | 10 | 0 |
| 6 |  | (0, 6, 7)<br>(0, 6, 18) | 16<br>8 | 2<br>2 |
| 7 |  | (2, 7, 19)<br>(2, 7, 20)<br>(6, 7, 2)<br>(6, 7, 19)<br>(6, 7, 20) | 8<br>8<br>8<br>8<br>8 | 4<br>4<br>4<br>4<br>4 |

Table 2: Continued.

| Intersections | The graphical representation of intersections and possible turnings | Possible turnings | Traffic capacity (hundreds of vehicles) | Intersection delay (min) |
|---|---|---|---|---|
| 8 |  | (9, 8, 14)<br>(9, 8, 16) | 8<br>6 | 2<br>2 |
| 9 |  | (1, 9, 8)<br>(1, 9, 10)<br>(1, 9, 13)<br>(10, 9, 8)<br>(10, 9, 13) | 12<br>12<br>8<br>8<br>8 | 4<br>4<br>4<br>4<br>4 |
| 10 |  | (9, 10, 11)<br>(9, 10, 12)<br>(11, 10, 12)<br>(11, 10, 9) | 10<br>10<br>8<br>10 | 3<br>3<br>3<br>3 |
| 11 |  | (2, 11, 10)<br>(2, 11, 15)<br>(10, 11, 15) | 12<br>12<br>12 | 2<br>2<br>2 |
| 12 |  | (10, 12, 13)<br>(10, 12, 15) | 12<br>12 | 2<br>2 |

second batch is saturated at this moment with the total route cost of 112, namely 11200 mins.

We repeat the above steps according to Table 5 and obtain the following results in Table 6.

*6.4. Case Analysis of One-to-Many Evacuation.* Simple case calculations are implemented in the above section for the one-to-one evacuation problem on the basis of MC-MFM. In this section, the form of evacuation is extended to one-to-many evacuation. According to the second ring road map of Beijing, the evacuation destinations include eight nodes, which are nodes 13, 14, 15, 16, 17, 18, 19, and 20. In specific evacuation, the problems of route optimization are ensured based on MFM.

Table 3: Traffic capacity and time schedule of each road section.

| Arc | Traffic capacity (hundreds of vehicles) | Travel time or delay (min) | Arc | Traffic capacity (hundreds of vehicles) | Travel time or delay (min) |
|---|---|---|---|---|---|
| (0, 1) | 20 | 2 | (0, 1, 9) | 16 | 0 |
| (0, 2) | 20 | 2.5 | (0, 2, 11) | 12 | 2 |
| (0, 6) | 20 | 3 | (0, 6, 7) | 16 | 0 |
| (1, 9) | 16 | 2 | (1, 9, 10) | 12 | 3 |
| (2, 11) | 18 | 2 | (1, 9, 13) | 8 | 3 |
| (6, 7) | 12 | 2.5 | (2, 11, 10) | 12 | 0 |
| (7, 2) | 10 | 3 | (6, 7, 2) | 8 | 0 |
| (9, 10) | 12 | 3 | (7, 2, 11) | 10 | 2 |
| (9, 13) | 16 | 3 | (9, 10, 12) | 10 | 3 |
| (10, 9) | 12 | 3 | (10, 9, 13) | 8 | 3 |
| (10, 12) | 16 | 1 | (10, 12, 13) | 12 | 0 |
| (11, 10) | 12 | 1.5 | (11, 10, 9) | 10 | 3 |
| (12, 13) | 10 | 4 | (11, 10, 12) | 8 | 3 |

Table 4: Optimal routes of one-to-one evacuation on MFM.

| Round | Route | Flow |
|---|---|---|
| $f_1 f_1$ | 0-1-9-10-12-13 | 8 |
| $f_2$ | 0-2-11-10-12-13 | 2 |
| $f_3$ | 0-2-11-10-9-13 | 8 |
| $f_4$ | 0-1-9-13 | 8 |
| $f_T$ | 4 | 26 |

To keep the comparability with the case in the previous calculation, the total number of vehicles in the source node is still 60. The numerical values of specific capacity and the amount of time consumed are shown in Table 1. Source node is node 0, and evacuation destinations are 13, 14, 15, 16, 17, 18, 19, and 20.

In spite of being a different form, we still use label censorship method for calculation. Specific calculation process is shown in Table 7.

As illustrated in Table 7, we can find the minimum cost route 0-6-18 with the cost of 8. As the bottleneck of traffic capacity on arc (0, 6, 18) is 8, this route is augmented chain and traffic flow should be increased with 8. After that, arc (0, 6, 18) in the first batch is saturated, and the total route cost is 64. In addition, in the process of finding the minimum cost routes in the next step, subsequent batches should be considered.

Similarly, route 0-1-4-16 in the first batch does not include the arc (0, 6, 18), and its cost is 10, which is the minimum cost in the remaining viable routes, so this route is augmented chain. Since the bottleneck of traffic flow is arc (4, 16), which is 10, traffic flow on this augmented chain can be increased with 10. At this point, arc (4, 16) in the second batch is saturated, and the total route cost is 100.

We repeat the above steps according to Table 7 and obtain the following results, see Table 8.

Compared with one-to-one evacuation, one-to-many evaluation will take just 13 minutes, which is much faster and more efficient. Therefore, we draw conclusions as follows: under normal circumstances, when a certain emergency occurs and the emergency evacuation is required, multiple destinations for the evacuation are quicker and more efficient than only one destination. Therefore, in order to prevent heavy casualties caused by major emergencies and improve evacuation efficiency, more evacuation roads and shelters should be built in the places with high population density and high occurrence of emergencies.

## 7. Conclusions

Appropriate traffic control measures must be taken into account in evacuation route optimization. Overall, this paper explores evacuation optimization and planning of evacuation routes, considering some traffic control measures. We abstract the road network as directive network, taking the vehicles waiting for evacuation in the source node as flows and evacuation road sections along the road as arcs which are under traffic control, such as one-way, two-way, delay, and so forth. Besides, the traffic capacity of intersections is expressed as the weights of network diagram. After model formulation, we use graph theory method to solve our model and then verify its feasibility.

Based on the case analysis to the second ring of Beijing, we present evacuation optimization and planning after emergencies happen. Although we make a lot of prerequisites and assumptions for modeling and calculation, there are still some issues remained to be explored as follows.

(1) Our paper considers the optimization of route under traffic control measures, but these measures like one-way, two-way, forbidding for passing, and intersection delay are finite. However, some measures with instant changes are not considered, and intersection delays are often changed in reality. All of the above factors are also required to be considered in future study.

Table 5: Minimum-cost emergency evacuation route.

| | | | | | |
|---|---|---|---|---|---|
| Minimum cost from the source node | 2 | 2 | 2.5 | 3 | 3 |
| Reviewed arc | (0, 1) | (0, 1, 9) | (0, 2) | (0, 6) | (0, 6, 7) |
| The cost of associated arc segments departing from the source node | (0, 1, 9), 2 | (1, 9), 4<br>(0, 1, 9), 7 | (0, 2, 11), 4.5 | (0, 6, 7), 3 | (6, 7), 5.5 |
| Minimum cost from the source node | 4 | 4.5 | 5.5 | 5.5 | 6.5 |
| Reviewed arc | (1, 9) | (0, 2, 11) | (6, 7) | (6, 7, 2) | (2, 11) |
| The cost of associated arc segments departing from the source node | (1, 9, 10), 7 | (2, 11), 6.5 | (6, 7, 2), 5.5 | (7, 2), 8.5 | (2, 11, 10), 6.5 |
| Minimum cost from the source node | 6.5 | 7 | 7 | 8 | 8.5 |
| Reviewed arc | (2, 11, 10) | (1, 9, 10) | (1, 9, 13) | (11, 10) | (7, 2) |
| The cost of associated arc segments departing from the source node | (11, 10), 8 | (9, 10), 10 | (9, 13), 10 | (11, 10, 9), 11<br>(11, 10, 12), 11 | (7, 2, 11), 10.5 |
| Minimum cost from the source node | 10 | 10 | 10 | 10.5 | 11 |
| Reviewed arc | (9, 10) | (9, 13) | (9, 10) | (7, 2, 11) | (11, 10, 12) |
| The cost of associated arc segments departing from the source node | (9, 10, 12), 13 | Reach the designated destination | (9, 10, 12), 13 | (2, 11), 12.5 | (10, 12), 12 |
| Minimum cost from the source node | 11 | 12 | 12 | 12.5 | 12.5 |
| Reviewed arc | (11, 10, 9) | (10, 12) | (10, 12, 13) | (2, 11) | (2, 11, 10) |
| The cost of associated arc segments departing from the source node | (10, 9), 14 | (10, 12, 13), 12 | (12, 13), 16 | (2, 11, 10), 12.5 | (11, 10), 14 |
| Minimum cost from the source node | 13 | 14 | 14 | 14 | 14 |
| Reviewed arc | (9, 10, 12) | (10, 12) | (10, 12, 13) | (10, 9) | (11, 10) |
| The cost of associated arc segments departing from the source node | (10, 12), 14 | (10, 12, 13), 14 | (12, 13), 18 | (10, 9, 13), 17 | (11, 10, 12), 17<br>(11, 10, 9), 17 |
| Minimum cost from the source node | 16 | 17 | 17 | 17 | 18 |
| Reviewed arc | (12, 13) | (10, 9, 13) | (11, 10, 9) | (11, 10, 12) | (10, 12) |
| The cost of associated arc segments departing from the source node | Reach the designated destination | (9, 13), 20 | (10, 9), 20 | (10, 12), 18 | (10, 12, 13), 18 |
| Minimum cost from the source node | 18 | 18 | 20 | 20 | 22 |
| Reviewed arc | (10, 12, 13) | (12, 13) | (9, 13) | (10, 9) | (12, 13) |
| The cost of associated arc segments departing from the source node | (12, 13), 22 | Reach the designated destination | Reach the designated destination | (10, 9, 13), 23 | Reach the designated destination |
| Minimum cost from the source node | 23 | 26 | | | |
| Reviewed arc | (10, 9, 13) | (9, 13) | | | |
| The cost of associated arc segments departing from the source node | (9, 13), 26 | Reach the designated destination | | | |

Table 6: Optimal routes of one-to-one evacuation on MC-MFM.

| Serial number | Route | Number of batch | Flow | Total time (minutes) |
|---|---|---|---|---|
| $f_1$ | 0-1-9-13 | 4 | 32 | 22 |
| $f_2$ | 0-2-11-10-12-13 | 2 | 16 | |
| $f_3$ | 0-1-9-10-12-13 | 2 | 8 | |
| $f_4$ | 0-2-11-10-9-13 | 1 | 4 | |
| $f_T$ | 4 | 9 | 60 | |

(2) We establish mathematical model and use graph theory, aiming at obtaining the maximum evacuation vehicle number with the minimum total time. In fact, we generally need to consider many factors in a variety of situations, and goals should not be single. In order to develop more realistic models to solve actual-world problems, we need to consider the integrated application of goal programming and graph theory when multiple goals should be achieved.

Table 7: Optimal routes of one-to-many emergency on MC-MFM.

| | | | | | |
|---|---|---|---|---|---|
| Minimum cost from the source node | 2 | 2.5 | 3 | 4 | 4 |
| Reviewed arc | (0, 1) | (0, 2) | (0, 6) | (0, 1, 4) | (0, 1, 9) |
| The cost of associated arc segments departing from the source node | (0, 1, 4), 4<br>(0, 1, 9), 4 | (0, 2, 3), 5.5 | (0, 6, 7), 5<br>(0, 6, 18), 5 | (1, 4), 6 | (1, 9), 6 |
| Minimum cost from the source node | 5 | 5 | 5.5 | 5.5 | 5.5 |
| Reviewed arc | (0, 6, 7) | (0, 6, 18) | (0, 2, 11) | (0, 2, 7) | (0, 2, 3) |
| The cost of associated arc segments departing from the source node | (6, 7), 7.5 | (6, 18), 8 | (2, 11), 7.5 | (2, 7), 8.5 | (2, 3), 7.5 |
| Minimum cost from the source node | 6 | 6 | 7.5 | 7.5 | 7.5 |
| Reviewed arc | (1, 4) | (1, 9) | (2, 3) | (2, 11) | (6, 7) |
| The cost of associated arc segments departing from the source node | (1, 4, 5), 8<br>(1, 4, 16), 8 | (1, 9, 8), 10<br>(1, 9, 10), 10<br>(1, 9, 13), 10 | (2, 3, 17), 9.5<br>(2, 3, 19), 9.5 | (2, 11, 10), 9.5<br>(2, 11, 15), 9.5 | (6, 7, 2), 11.5<br>(6, 7, 19), 11.5<br>(6, 7, 20), 11.5 |
| Minimum cost from the source node | 8 | 8 | 8 | 8.5 | 9.5 |
| Reviewed arc | (6, 18) | (1, 4, 5) | (1, 4, 16) | (2, 7) | (2, 3, 17) |
| The cost of associated arc segments departing from the source node | Reach the destination | (4, 5), 11 | (4, 16), 10 | (2, 7, 19), 12.5<br>(2, 7, 20), 12.5 | (3, 17), 10.5 |
| Minimum cost from the source node | 9.5 | 9.5 | 9.5 | 10 | 10 |
| Reviewed arc | (2, 3, 19) | (2, 11, 10) | (2, 11, 15) | (1, 9, 8) | (1, 9, 13) |
| The cost of associated arc segments departing from the source node | (2, 19), 12.5 | (11, 10), 11 | (11, 15), 10.5 | (9, 8), 11 | (9, 13), 13 |
| Minimum cost from the source node | 10 | 10.5 | 10.5 | 11 | 11 |
| Reviewed arc | (4, 16) | (3, 17) | (11, 15) | (4, 5) | (4, 5, 18) |
| The cost of associated arc segments departing from the source node | Reach the destination | Reach the destination | Reach the destination | (4, 5, 18), 11 | (5, 18), 13 |
| Minimum cost from the source node | 11 | 11 | 11.5 | 11.5 | 11.5 |
| Reviewed arc | (11, 10) | (9, 8) | (6, 7, 19) | (6, 7, 2) | (6, 7, 20) |
| The cost of associated arc segments departing from the source node | (11, 10, 9), 14<br>(10, 10, 12), 14 | (9, 8, 14), 13<br>(9, 8, 16), 13 | (7, 19), 13.5 | (7, 2), 14.5 | (7, 20), 12.5 |
| Minimum cost from the source node | 12.5 | 12.5 | 12.5 | 12.5 | 13 |
| Reviewed arc | (3, 19) | (2, 7, 19) | (2, 7, 20) | (7, 20) | (9, 13) |
| The cost of associated arc segments departing from the source node | Reach the destination | (7, 19), 14.5 | (7, 20), 13.5 | Reach the destination | Reach the destination |

Table 8: Optimal routes of one-to-many evacuations on MC-MFM.

| Serial number | Route | Number of batch | Flow | Total time (minutes) |
|---|---|---|---|---|
| $f_1$ | 0-6-18 | 2 | 16 | 13 |
| $f_2$ | 0-1-4-16 | 1 | 10 | |
| $f_3$ | 0-2-3-17 | 1 | 10 | |
| $f_4$ | 0-2-11-15 | 1 | | |
| $f_5$ | 0-6-7-20 | 1 | 8 | |
| $f_6$ | 0-1-9-13 | 1 | 6 | |
| $f_T$ | 5 | 7 | 60 | |

## Conflict of Interests

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

## Acknowledgments

This research was supported by the National Natural Science Foundation of China (nos. 71102174, 71372019, and 71173017), Beijing Natural Science Foundation of China (no. 9123028), Beijing Philosophy & Social Science Foundation of China (no. 11JGC106), Beijing Higher Education Young Elite Teacher Project (no. YETP1173) Specialized Research Fund for Doctoral Program of Higher Education of China (no. 20111101120019), Program for New Century Excellent Talents in University of China (nos. NCET-10-0048, NCET-10-0043), Key Project Cultivation Fund of the Scientific and Technical Innovation Program in Beijing Institute of Technology of China (no. 2011DX01001), Excellent Young Teacher in Beijing Institute of Technology of China (no. 2010YC1307), and Basic Research in Beijing Institute of Technology of China (no. 20102142013).

## References

[1] C. E. Haque, "Perspectives of natural disasters in East and South Asia, and the Pacific Island States: socio-economic correlates and needs assessment," *Natural Hazards*, vol. 29, no. 3, pp. 465–483, 2003.

[2] J. P. Yuan, Z. Fang, Z. M. Lu, and H. C. Huang, "Discussion of large-scale personnel emergency evacuation when the city is hit by natural adversity," *Natural Disaster Journal*, vol. 14, no. 6, pp. 116–119, 2005.

[3] M. K. Lindell and C. S. Prater, "Critical behavioral assumptions in evacuation time estimate analysis for private vehicles: examples from hurricane research and planning," *Journal of Urban Planning and Development*, vol. 133, no. 1, pp. 18–29, 2007.

[4] M. X. Gao, *The Research of Evacuation Organization Measures Considering the Effect of Traffic Management and Control Measures*, Tianjing University, 2008.

[5] S. Kim, S. Shekhar, and M. Min, "contraflow transportation network reconfiguration for evacuation route planning," *Knowledge and Data Engineering, IEEE Transactions*, vol. 20, no. 8, pp. 1115–1129, 2008.

[6] Y. Liu, X. Lai, and G. L. Chang, "Cell-based network optimization model for staged evacuation planning under emergencies," *Transportation Research Record*, vol. 1964, no. 1, pp. 127–135, 2006.

[7] G. Lämmel, M. Rieser, K. Nagel et al., *Emergency Preparedness in the Case of a Tsunami-Evacuation Analysis and traffic optimization for the Indonesian City of Padang*, Springer, Berlin, Germany, 2010.

[8] T. J. Cova and J. P. Johnson, "A network flow model for lane-based evacuation routing," *Transportation Research Part A*, vol. 37, no. 7, pp. 579–604, 2003.

[9] S. Mamada, K. Makino, and S. Fujishige, "Evacuation problems and dynamic network flows," in *Proceedings of the SICE Annual Conference*, pp. 2373–2378, Sapporo, Japan, August 2004.

[10] Y.-C. Chiu, H. Zheng, J. Villalobos, and B. Gautam, "Modeling no-notice mass evacuation using a dynamic traffic flow optimization model," *IIE Transactions*, vol. 39, no. 1, pp. 83–94, 2007.

[11] W. Choi, H. W. Hamacher, and S. Tufekci, "Modeling of building evacuation problems by network flows with side constraints," *European Journal of Operational Research*, vol. 35, no. 1, pp. 98–110, 1988.

[12] A. F. Han, "TEVACS. Decision support system for evacuation planning in Taiwan," *Journal of Transportation Engineering*, vol. 116, no. 6, pp. 821–830, 1990.

[13] B. Hoppe and E. Tardos, "Polynomial time algorithms for some evacuation problems," in *Proceedings of the 5th Annual SIAM Symposium on Discrete Algorithms*, pp. 433–441, January 1994.

[14] H. Klüpfel, T. Meyer-König, J. Wahle et al., "Microscopic simulation of evacuation processes on passenger ships," in *Theory and Practical Issues on Cellular Auto-Mana*, pp. 63–71, Springer, London, UK, 2001.

[15] G. Theodoulou and B. Wolshon, "Alternative methods to increase the effectiveness of freeway contraflowevacuation," *Transportation Research Record*, no. 1865, pp. 48–56, 2004.

[16] S. E. Adewumi and E. J. D. Garba, "An algorithm for encrypting messages using matrix inversion," *The Science World JOURNAL*, vol. 2, no. 4, pp. 13–14, 2007.

[17] A. Stepanov and J. M. Smith, "Multi-objective evacuation routing in transportation networks," *European Journal of Operational Research*, vol. 198, no. 2, pp. 435–446, 2009.

[18] R. Pal and I. Bose, "An optimization based approach for deployment of roadway incident response vehicles with reliability constraints," *European Journal of Operational Research*, vol. 198, no. 2, pp. 452–463, 2009.

[19] O. M. Saeed and B. Ram, "Two-phase evacuation route planning approach using combined path networks for buildings and roads," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 233–245, 2013.

[20] W. Zhang, W. Guan, J. Ma et al., "Nonlinear min-cost-pursued route-swapping dynamic system," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 162128, 10 pages, 2013.

[21] C. E. Dunn and D. Newton, "Optimal routes in GIS and emergency planning applications," *Area*, vol. 24, no. 3, pp. 259–267, 1992.

[22] T. Yamada, "A network flow approach to a city emergency evacuation planning," *International Journal of Systems Science*, vol. 27, no. 10, pp. 931–936, 1996.

[23] V. B. G. Campos and P. A. L. da Silva, "Netto Evacuation transportation planning: a method of identify optimal independent routes," in *Proceedings of the International Conference on Urban Transport*, pp. 555–564, 2000.

[24] V. V. Dixit and E. Radwan, "Hurricane evacuation: origin, route, and destination," *Journal of Transportation Safety & Security*, vol. 1, no. 1, pp. 74–84, 2009.

[25] T. Koike, T. Kanamori, and Y. Sato, "An evacuation plan from a fire following earthquake in a congested city area," *Journal of Earthquake and Tsunami*, vol. 4, no. 1, pp. 33–49, 2010.

[26] P. B. Luh, C. T. Wilkie, S. C. Chang et al., "Modeling and optimization of building emergency evacuation considering blocking effects on crowd movement," *Automation Science and Engineering, IEEE Transactions*, vol. 9, no. 99, p. 1, 2012.

[27] H. W. Hamacher and S. A. Tjandra, "Mathematical modeling of evacuation problems-a state of the art," in *Pedestrian and Evacuation Dynamics*, pp. 227–266, 2002.

[28] S. Mamada, K. Makino, T. Takabatake et al., "The evacuation problem, dynamic network flows, and algorithms," in *Proceedings of the IEEE Annual Conference (SICE '03)*, vol. 3, pp. 2807–2811, 2003.

[29] Q. Lu, B. George, and S. Shekhar, "Capacity constrained routing algorithms for evacuation planning: a summary of results," in *Advances in Spatial and Temporal Databases*, pp. 291–307, Springer, Berlin, Germany, 2005.

[30] P. Lin, *A Dynamic Network Flow Optimization for Large Scale Emergency Evacuation*, City university of Hong Kong, 2006.

[31] G. Fuellerer, K. F. Doerner, R. F. Hartl, and M. Iori, "Ant colony optimization for the two-dimensional loading vehicle routing problem," *Computers and Operations Research*, vol. 36, no. 3, pp. 655–673, 2009.

[32] S. C. H. Leung, X. Zhou, D. Zhang, and J. Zheng, "Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem," *Computers and Operations Research*, vol. 38, no. 1, pp. 205–215, 2011.

[33] V. Pillac, M. Gendreau, C. Guéret et al., "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, no. 1, pp. 1–11, 2012.

[34] W. Xu, L. Miao, and W. H. Lin, "Stochastic user equilibrium assignment in schedule-based transit networks with capacity constraints," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 910754, 15 pages, 2012.

[35] W. C. Grimmell and N. S. V. Rao, "On source-based route computation for quickest paths under dynamic bandwidth

constraints," *International Journal of Foundations of Computer Science*, vol. 14, no. 3, pp. 503–523, 2003.

[36] M. Murtaza, M. Welzl, and B. Yener, "On the accurate identification of network paths having a common bottleneck," in *Proceedings of the (SIGCOMM '08)*, pp. 17–22, ACM, Seattle, Wash, USA, 2008.

[37] L. Deng, W. Gao, Y. Fu et al., "Optimal design of the feeder-bus network based on the transfer system," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 483682, 10 pages, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 894573, 12 pages
http://dx.doi.org/10.1155/2014/894573

*Research Article*

# A Collaborative Scheduling Model for the Supply-Hub with Multiple Suppliers and Multiple Manufacturers

**Guo Li,[1] Fei Lv,[2] and Xu Guan[3]**

[1] *School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China*
[2] *School of Management, Huazhong University of Science and Technology, Wuhan 430074, China*
[3] *Economics and Management School, Wuhan University, Wuhan 430072, China*

Correspondence should be addressed to Xu Guan; dageguan@gmail.com

Received 1 October 2013; Accepted 17 November 2013; Published 15 April 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

This paper investigates a collaborative scheduling model in the assembly system, wherein multiple suppliers have to deliver their components to the multiple manufacturers under the operation of Supply-Hub. We first develop two different scenarios to examine the impact of Supply-Hub. One is that suppliers and manufacturers make their decisions separately, and the other is that the Supply-Hub makes joint decisions with collaborative scheduling. The results show that our scheduling model with the Supply-Hub is a NP-complete problem, therefore, we propose an auto-adapted differential evolution algorithm to solve this problem. Moreover, we illustrate that the performance of collaborative scheduling by the Supply-Hub is superior to separate decision made by each manufacturer and supplier. Furthermore, we also show that the algorithm proposed has good convergence and reliability, which can be applicable to more complicated supply chain environment.

## 1. Introduction

To avoid the supply delay risk caused by any supplier, in practice, manufacturer/assembler normally prefers to outsource its purchasing business to the third party logistics (TPL) and requires his suppliers to hold inventory in the warehouse operated by TPL. Given this trend, a new type of TPL arises in recent years, which is called Supply-Hub. Investigated by many scholars [1–3], the Supply-Hub is an integrated logistics provider with a series of logistic services (e.g., assemblage, distribution, and warehouse), which is widely applied in the auto and electronic industries to support the manufacturer to implement the JIT (Just In Time) production, so as to respond to the market changes rapidly. For example, BAX GLOBAL and UPS are two typical logistic companies operated under Supply-Hub mode with Chinese auto companies.

For most Supply-Hubs, they are located near the manufacturer's factory, so as to store most of the raw materials delivered by the suppliers. According to the agreement, the Supply-Hub will charge the suppliers for the components consumed during a fixed period of time. However, during this process, it is very hard for the Supply-Hub to coordinate the production and the delivery among different suppliers, specific to how to precisely determine the supplier's production lot, the distribution frequency, and the distribution quantity. In actual business activities, when the material flows are sufficiently large, the coordination and optimization of production and distribution based on the Supply-Hub can bring substantial benefits to the members in the supply chain.

Our paper is related to the vast literature, which can be divided into two groups. The first group concerns the order allocation, vehicle routing, and production planning. Hahm and Yano [4] explore the economic lot and delivery scheduling problem. Moreover, Hahm and Yano [5, 6], Khouja [7], and Clausen and Ju [8] consider the problems of determining the production scheduling and distribution intervals for different types of components when a supplier provides different kinds of components. Vergara et al. [9] propose the genetic algorithm to make production scheduling and cycle time arrangements for many kinds of components in a simple multistage supply chain, where each supplier provides one or a variety of products for the upstream supplier or assembler.

Khouja [10] examines production sequencing and distribution scheduling in a single-product and multi-product supply chain when production intervals equal distribution intervals. However, all of the above literature focus on a simple supply chain structure, and the problems of production and distribution in an assembly supply chain with multiple suppliers and multiple manufacturers have not been involved. Pundoor [11] first establishes a cooperative scheduling model of production and distribution in a multi-suppliers, one-warehouse, and one-customer system. The supplier's production and distribution interval and warehouse's distribution interval are collaboratively optimized to minimize the unit production and logistics cost in the upstream supply chain wherein supplier's production capability is limited. However, the model does not consider the transportation constraint from the warehouse to the manufacturer. Torabi et al. [12] investigate the lot and delivery scheduling problem in a simple supply chain where a single supplier produces multiple components on a flexible flow line (FFL) and delivers them directly to an assembly facility (AF). They also develop a new mixed integer nonlinear program (MINLP) and an optimal enumeration method to solve the problem. Naso et al. [13] focus on the ready-mixed concrete delivery. They propose a novel meta-heuristic approach based on a hybrid genetic algorithm combined with constructive heuristics. Ma and Gong [14] extend the work of Pundoor [11] to a multi-suppliers and one-manufacturer system based on the Supply-Hub. In their model the production lot and distribution interval are optimized from either supplier's or manufacturer's perspective.

The second group is about the coordination with the application of Supply-Hub. Barnes et al. [1] find that the Supply-Hub is an innovative strategy to reduce cost and improve responsiveness. They further define the concept of Supply-Hub and review its development by analyzing the practical case. They also propose a prerequisite of establishing Supply-Hub and the main way of operating Supply-Hub. Shah and Goh [3] explore the operation strategy of Supply-Hub to achieve the joint operation management between customers and upstream suppliers. Moreover, they analyze how to manage the supply chain better in a vendor-managed inventory model. Furthermore, they find that the relationships between the operation strategies and performance evaluations of Supply-Hub are complex and nonlinear. As a result, they propose a hierarchical structure to help the Supply-Hub achieve the balance among different members. Lin and Chen [15] propose a generalized hub-and-spoke network in a capacitated and directed network configuration that integrates the operations of three common hub-and-spoke networks: pure, stopover, and center directs. They also develop an implicit enumeration algorithm with embedded integrally constrained multicommodity min-cost flow. Lin [16] studies the integrated hierarchical hub-and-spoke network design problem for dual services. They propose a directed network configuration and formulate a link-based integer mathematical model, and also develop a link-based implicit enumeration with an embedded degree and time constrained spanning tree algorithm. Charles et al. [17] investigate how implement integrated logistics hubs by considering six independent industrial sectors with reference models and systems. The research results provide a field tested method for deriving integrated logistics hub models in different manufacturing economies with notes that provide sufficient methodological details for repeating the construction of logistics hubs in other manufacturing economies.

Based on the Supply-Hub, Ma and Gong [14] develop collaborative decision-making models of production and distribution considering the matching of distribution quantity between suppliers. The result shows that the total supply chain cost and the production cost of suppliers decrease significantly, but the logistics cost of manufacturers and the operational cost of Supply-Hub increase. In order to explore the effect of the supply chain design caused by the structural changes in the assembly system, Li et al. [18] establish several supply chain design models (one without supply center, one with single-stage supply center, and one with two-stage supply center) according to the characteristics of bill of material (BOM) and the relationships of multiple properties among suppliers. With the consideration that multiple suppliers provide different components to a manufacturer based on the Supply-Hub, Gui and Ma [19] establish an economical order quantity model in such two ways as picking up separately from different suppliers and milk-run picking up. The result shows that the sensitivity to carriage quantity of the transportation cost and the demand variance in different components have an influence on the choices of two picking up ways. Li et al. [20] give a thorough review about collaborative operation and optimization in supply logistics based on the Supply-Hub and point out that how to coordinate suppliers and share risks is still to be explored.

Obviously, the above literature does not take the coordination issue of Supply-Hub into account. In the actual operation process of Supply-Hub, the service for multiple suppliers and multiple manufacturers are often provided by a single Supply-Hub. For example, BAX GLOBAL takes in charge of the logistics services in Southeast Asia for Apple, Dell, and IBM. From the perspective of Supply-Hub, how to integrate resources of multiple suppliers and multiple manufacturers is the key point of implementing JIT policy in the supply chain.

## 2. Problem Definition and Notation

*2.1. Problem Description.* Let us consider the following operation process: each manufacturer sends its material requirement plan to the Supply-Hub and corresponding suppliers based on a rolling plan. After that, the Supply-Hub optimizes and arranges the production and distribution activities for each supplier based on the information of production costs and inventory status. Finally, the Supply-Hub implements JIT direct-station distribution according to material requirement plan in each week or day provided by each manufacturer. The illustration of the process is shown in Figure 1. It is worth mentioning that the production information is freely shared among the suppliers, the Supply-Hub and the manufacturers.

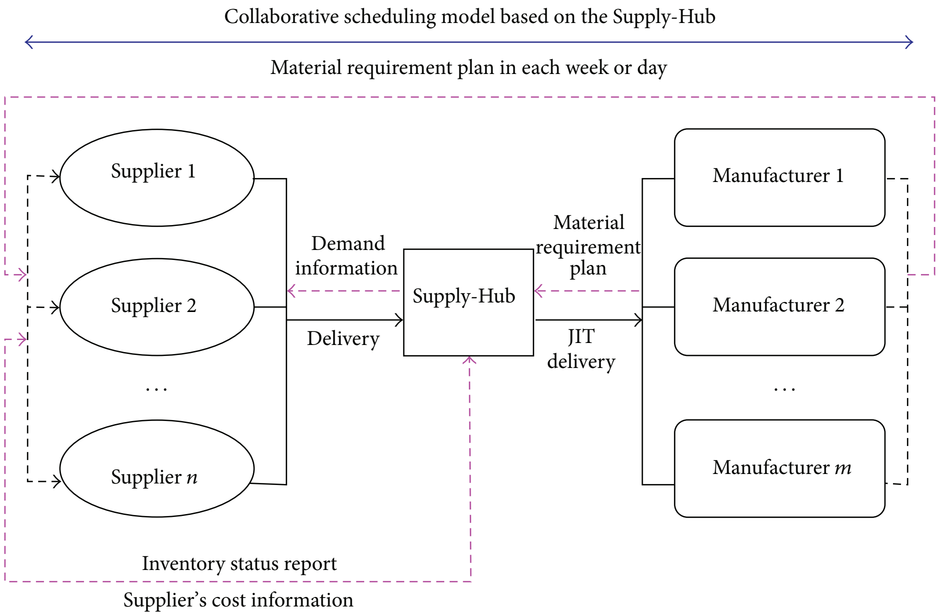

Collaborative scheduling model based on the Supply-Hub
Material requirement plan in each week or day
Supplier 1
Supplier 2
...
Supplier n
Demand information
Delivery
Supply-Hub
Material requirement plan
JIT delivery
Manufacturer 1
Manufacturer 2
...
Manufacturer m
Inventory status report
Supplier's cost information

Figure 1: Multi-suppliers and multi-manufacturers operation mode based on the Supply-Hub.

Note that the coordination scheduling is to implement the JIT distribution of components required by each manufacturer with minimal cost. To achieve this goal, the manufacturer's distribution lot, the supplier's production lot, and the distribution frequency should be optimized through integration of the entire supply chain and logistics operation based on the Supply-Hub. In Figure 1, the Supply-Hub provides the service for $m$ manufacturers and $n$ suppliers.

For manufacturer $j$, where $j = 1, 2, \ldots, m$, the number of its suppliers is $k_j$, where $1 \le k_j \le n$. It indicates that the components required by manufacturer $j$ are provided by $k_j$ suppliers. For a certain supplier $i$, where $i = 1, 2, \ldots, n$, the number of components required by manufacturer is $l_i$, where $1 \le l_i \le m$. It implies that the components provided by supplier $i$ are required by $l_i$ manufacturer. Therefore, the multi-suppliers and multi-manufacturers system based on the Supply-Hub considered in our paper is more universal and versatile.

*2.2. Assumptions and Notations.* The Supply-Hub takes charge in the components purchasing and JIT direct-station distribution for $m$ manufacturers. Component $i$ required by manufacturer $j$ is delivered to manufacturer $j$ by the Supply-Hub at suitable interval $R_{h,j}$, and component $i$ from supplier $i$ was delivered to the Supply-Hub at regular interval $R_{ij}$. According to the distribution lot to the Supply-Hub, the purchasing lot is determined by supplier $i$.

Define

$$w_{ij} = \begin{cases} 1 & \text{if supplier } i \text{ provides component } i \text{ for manufacturer } j \\ 0 & \text{else.} \end{cases} \quad (1)$$

*2.2.1. Assumptions.* The specific assumptions are as follows.

(1) Each supplier provides one kind of the component for a manufacturer, and demand for the component is constant. Note that our results remain unchanged if a certain supplier can provide a variety of components, since it can be actually converted to multiple suppliers and each provides one component.

(2) The transportation cost of component $i$ required by manufacturer $j$ from supplier $i$ to the Supply-Hub is composed of a fixed cost $F_{ij}$ and a variable cost $V_{ij}$, and the transportation cost from the Supply-Hub to manufacturer $j$ also contains a fixed cost $F_{h,ij}$ and a variable cost $V_{h,ij}$.

(3) The lead time for each level of the supply chain is constant, and it is assumed to be zero without loss of generality.

(4) Shortages are not allowed.

(5) Time horizon is infinite.

*2.2.2. Notations.* The input parameters and decision variables for manufacturers, the Supply-Hub, and suppliers, are denoted by the subscripts $m$, $h$, and $s$, respectively.

*Manufacturers:*

$m$ is the number of manufacturer; where $j = 1, 2, \ldots, m$

$d_{ij}$ is the annual demand of manufacturer $j$ for the component $i$ (units/year);

$h_{m,ij}$ is the manufacturer $j$'s holding cost per unit per year for component $i$;

$A_{m,j}$ is the order cost for manufacturer $j$ (\$);

$T_j$ is the cycle time (year).

*The Supply-Hub:*

$A_h$ is the fixed-order/setup cost per cycle for the Supply-Hub;

$h_{h,i}$ is the Supply-Hub's holding cost per unit per year for component $i$;

$M_{h,j}$ is an integer multiplier to adjust the order quantity of the Supply-Hub to that of manufacturer $j$;

$F_{h,j}$ is the fixed transportation cost from the Supply-Hub to manufacturer $j$;

$V_{h,j}$ is the variable transportation cost from the Supply-Hub to manufacturer $j$.

*Suppliers:*

$n$ is the number of suppliers, where $i = 1, 2, \ldots, n$;

$A_{s,i}$ is the order cost for supplier $i$;

$h_{s,i}$ is the supplier's holding cost per unit per unit per year for component $i$;

$M_{s,ij}$ is an integer multiplier to adjust the order quantity of the supplier $i$ whose component is required by manufacturer $j$ to that of the Supply-Hub;

$F_{s,ij}$ is the fixed transportation cost for component $i$ required by manufacturer $j$ from supplier $i$ to the Supply-Hub;

$V_{s,ij}$ is the variable transportation cost for component $i$ required by manufacturer $j$ from supplier $i$ to the Supply-Hub.

## 3. Model Formulation

*3.1. Manufacturer's Cost Function.* Manufacturer $j$ orders $\sum_{i=1}^{n} d_{ij}T_j$ units from the Supply-Hub every $T_j$. The total annual cost for a manufacturer is the sum of the annual order cost, $A_{m,j}/T_j$, and the annual holding cost, $\sum_{i=1}^{n} d_{ij}T_j h_{m,ij}/2$. The annual cost function for manufacturer $j$ is given by

$$C_{m,j} = \frac{A_{m,j}}{T_j} + \frac{\sum_{i=1}^{n} d_{ij}T_j h_{m,ij}}{2}. \qquad (2)$$

The annual manufacturers' cost is the sum of $C_{m,j}$ for $m$ manufacturers, and it is given as

$$C_m = \sum_{j=1}^{m} C_{m,j}\left(T_j\right) = \sum_{j=1}^{m}\left(\frac{A_{m,j}}{T_j} + \frac{\sum_{i=1}^{n} d_{ij}T_j h_{m,ij}}{2}\right), \qquad (3)$$

where $T_j$ is a decision variable in (3), and the optimal cycle time for manufacturer $j$ is

$$T_j^* = \sqrt{\frac{2A_{m,j}}{\sum_{i=1}^{n} d_{ij}h_{m,ij}}}. \qquad (4)$$

In this paper, an optimal value of decision variable will be indicated by an asterisk $(*)$.

*3.2. Supply-Hub's Cost Function.* The Supply-Hub manages its upstream manufacturers separately; thus, it places an order for manufacturer $j$ every $M_{h,j}T_j$ and transports the components to manufacturer $j$ every $T_j$. The Supply-Hub's annual cost to satisfy the demand of manufacturer $j$ is

$$C_{h,j}\left(M_{h,j}\right) = \frac{A_h}{M_{h,j}T_j} + \sum_{i=1}^{n}\left[\frac{h_{h,i}}{2}\left(M_{h,j}-1\right)d_{ij}T_j\right] + \frac{F_{h,j} + \sum_{i=1}^{n} d_{ij}T_j V_{h,j}}{T_j}, \qquad (5)$$

where the terms $A_h/M_{h,j}T_j$, $\sum_{i=1}^{n}[(h_{h,i}/2)(M_{h,j}-1)d_{ij}T_j]$, and $(F_{h,j} + \sum_{i=1}^{n} d_{ij}T_j V_{h,j})/T_j$ are the Supply-Hub's annual order cost, the holding cost, and transportation cost for manufacturer $m$ which requires component $i$ from $n$ suppliers. Then the Supply-Hub's total cost is the sum of (5) for $m$ manufacturers, and it is given as

$$\begin{aligned} C_h &= \sum_{j=1}^{m} C_{h,j}\left(M_{h,j}\right) \\ &= \sum_{j=1}^{m}\left\{\frac{A_h}{M_{h,j}T_j} + \sum_{i=1}^{n}\left[\frac{h_{h,i}}{2}\left(M_{h,j}-1\right)d_{ij}T_j\right] + \frac{F_{h,j} + \sum_{i=1}^{n} d_{ij}T_j V_{h,j}}{T_j}\right\}, \end{aligned} \qquad (6)$$

where $M_{h,j}$ is a decision variable in (6), and the Supply-Hub's optimal cycle time for manufacturer $j$ is

$$M_{h,j} = \frac{1}{T_j}\sqrt{\frac{2A_h}{\sum_{i=1}^{n} h_{h,i}d_{ij}}}. \qquad (7)$$

*3.3. Supplier's Cost Function.* The Supply-Hub has $n$ suppliers to provide all $n$ components. When manufacturer $j$ places an order of size $\sum_{i=1}^{n} d_{ij}T_j$ with the Supply-Hub every $T_j$ and as discussed above, the Supply-Hub determines its order quantity $M_{h,j}d_{ij}T_j$ for the supplier $i$. In order to fulfill the demand of manufacturer $j$, the order of size $M_{h,j}d_{ij}T_j$ will be placed by the Supply-Hub, and shipment will occur every $M_{h,j}T_j$. The annual cost for supplier $i$ is written as

$$C_{s,i}\left(M_{s,ij}\right) = \sum_{j=1}^{m}\left[\frac{A_{s,i}}{M_{s,ij}M_{h,j}T_j} + \frac{h_{s,i}M_{h,j}}{2}\left(M_{s,ij}-1\right)d_{ij}T_j + \frac{F_{s,ij} + V_{s,ij}M_{h,j}T_j d_{ij}}{M_{h,j}T_j}\right], \qquad (8)$$

where the terms $A_{s,i}/M_{s,ij}M_{h,j}T_j$, $(h_{s,i}M_{h,j}/2)(M_{s,ij}-1)d_{ij}T_{ij}$, and $(F_{s,ij}+V_{s,ij}M_{h,j}T_j d_{ij})/M_{h,j}T_j$ are, respectively, the annual order cost, holding cost, and transportation cost for supplier $i$ to meet the annual demand for components required by the Supply-Hub. Then the collective annual cost for $n$ suppliers is given as

$$\begin{aligned} C_s &= \sum_{i=1}^{n} C_{s,i}\left(M_{s,ij}\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\left[\frac{A_{s,i}}{M_{s,ij}M_{h,j}T_j} + \frac{h_{s,i}M_{h,j}}{2}\left(M_{s,ij}-1\right)d_{ij}T_j \right. \\ &\qquad \left. + \frac{F_{s,ij}+V_{s,ij}M_{h,j}T_jd_{ij}}{M_{h,j}T_j}\right], \end{aligned} \tag{9}$$

where $M_{s,ij}$ is a decision variable in (9), and the supplier $i$'s optimal cycle time for manufacturer $j$ is

$$M_{s,ij} = \frac{1}{M_{h,j}T_j}\sqrt{\frac{2A_{s,i}}{d_{ij}}}. \tag{10}$$

*3.4. Solution Procedures with Decentralized Decision*

(1) Each manufacturer $j$ determines its optimal cycle time, $T_j^* = \sqrt{2A_{m,j}/\sum_{i=1}^{n} d_{ij}h_{m,ij}}$, where $i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$. Then the collective annual manufacturers' cost $C_m$ is computed from (3).

(2) The value of $T_j^*$ is input into (7), $M_{h,j} = (1/T_j^*)\sqrt{2A_h/\sum_{i=1}^{n} h_{h,i}d_{ij}}$. If $C_{h,j}(\lceil M_{h,j}\rceil) \geq C_{h,j}(\lfloor M_{h,j}\rfloor)$, then $M_{h,j}^* = \lfloor M_{h,j}\rfloor$. Or else $M_{h,j}^* = \lceil M_{h,j}\rceil$. This should be repeated for $m$ manufacturers, after which the collective Supply-Hub's annual cost, $C_h = \sum_{j=1}^{m} C_{h,j}$, is computed from (6).

(3) The values of $T_j^*$ and $M_{h,j}^*$ are input into (8), and (8) is minimized by searching the optimal value of $M_{s,ij}$. If $C_{s,i}(\lceil M_{s,ij}\rceil) \geq C_{s,i}(\lfloor M_{s,ij}\rfloor)$, then $M_{s,ij}^* = \lfloor M_{s,ij}\rfloor$, or else $M_{s,ij}^* = \lceil M_{s,ij}\rceil$. This may be repeated for $m \cdot n$ times because the component $i$ provided by supplier $i$ may be required by manufacturer $j$, where $i = 1, 2, \ldots, n, j = 1, 2, \ldots, m$. Then the collective supplier's annual cost $C_s$ is computed from (9).

(4) The value of optimal $T_j^*$, $M_{h,j}^*$, and $M_{s,ij}^*$ for each side should be recorded and the total supply chain cost for the case of no coordination is $C_{nsc} = C_m + C_h + C_s$, which can be obtained after the above three steps.

## 4. Supply Chain Coordination

The annual supply chain's cost is determined by summing (3), (6), and (9) to obtain

$$\begin{aligned} C_{\text{csc}} &= C_m + C_h + C_s \\ &= \sum_{j=1}^{m}\left(\frac{A_{m,j}}{T_j} + \frac{\sum_{i=1}^{n} d_{ij}T_jh_{m,ij}}{2}\right) \\ &\quad + \sum_{j=1}^{m}\left\{\frac{A_h}{M_{h,j}T_j} + \sum_{i=1}^{n}\left[\frac{h_{h,i}}{2}\left(M_{h,j}-1\right)d_{ij}T_j\right] \right. \\ &\qquad \left. + \frac{F_{h,j}+\sum_{i=1}^{n} d_{ij}T_jV_{h,j}}{T_j}\right\} \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{m}\left[\frac{A_{s,i}}{M_{s,ij}M_{h,j}T_j} + \frac{h_{s,i}M_{h,j}}{2}\left(M_{s,ij}-1\right)d_{ij}T_j \right. \\ &\qquad \left. + \frac{F_{s,ij}+V_{s,ij}M_{h,j}T_jd_{ij}}{M_{h,j}T_j}\right]. \end{aligned} \tag{11}$$

This is a centralized decision-making process, in which the Supply-Hub tries to schedule and optimize each decision variable for the entire supply chain. It is general and practical that the Supply-Hub takes charge of distribution frequency and purchasing frequency for the suppliers and the manufacturers, respectively.

Note that (11) is convex and differentiable over $T_j$, where $\partial^2 C_{\text{csc}}/\partial^2 T_j = \sum_{j=1}^{m}[(2/T_j^3)(A_{m,j} + (A_h/M_{h,j}) + F_{h,j} + \sum_{i=1}^{n}(A_{s,i}/M_{s,ij}M_{h,j}) + \sum_{i=1}^{n}(F_{s,ij}/M_{h,j}))] > 0$ for every $T_j > 0$, since $A_{m,j}$, $A_h$, $F_{h,j}$, $F_{s,ij}$, $A_{s,i}$, $M_{h,j}$, $M_{s,ij} > 0$. Therefore at a particular set of values for $M_{s,ij} \geq 1$ and $M_{h,j} \geq 1$, where $M_{s,ij}$ and $M_{h,j}$ are integer, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, the first derivative of (11) should be set to zero and the optimal $T_j^*$ was obtained. Consider

$$\begin{aligned} T_j^* = \Bigg( & 2\left[A_{m,j} + \frac{A_h}{M_{h,j}} + F_{h,j} + \sum_{i=1}^{n}\frac{A_{s,i}}{\left(M_{s,ij}M_{h,j}\right)} + \sum_{i=1}^{n}\frac{F_{s,ij}}{M_{h,j}}\right] \\ & \times \left(\sum_{i=1}^{n} d_{ij}h_{m,ij} + \left(M_{h,j}-1\right)\sum_{i=1}^{n} h_{h,i}d_{ij} + M_{h,j} \right. \\ & \left. \times \sum_{i=1}^{n}\left(M_{s,ij}-1\right)d_{ij}\right)^{-1}\Bigg)^{1/2}. \end{aligned} \tag{12}$$

*4.1. Complexity Analysis for This Problem.* The complexities of solving this problem are analyzed as follows. The optimal $T_j$, $M_{h,j}$, and $M_{s,ij}$ should be obtained to minimize the supply chain's cost $C_{\text{csc}}$, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. If a certain group of solution to this problem was proved NP-complete, then the whole group of solutions to this problem must be NP-complete.

Taking supplier 1 as the representative case, whose problem is to minimize $C_{\text{csc}}(M_{s,1j}, M_{h,j}, T_j \mid j = 1, 2, \ldots, m)$. We define this problem as $P$. If the problem $P$ can be proved to equal partition problem, then the problem $P$ is NP-complete.

*Partition Problem.* Given the positive integer $n$, $B$, and a group of positive integers $G = \{x_1, x_2, \ldots, x_n\}$, then $\sum_{i=1}^{n} x_i = 2B$, can $G$ be divided into group $G_1$ and $G - G_1$ to make $\sum_{x_i \in G_i} x_i = \sum_{x_i \in G - G_i} x_i = B$.

**Lemma 1.** *Partition is NP-complete; see Garey and Johnson [21].*

**Proposition 2.** *The problem P is NP-complete.*

*Proof.* We should transform the problem $P$ to partition. Let the sets $T$, $M_h$, $M_{s,1}$, with $|T| = |M_h| = |M_{s,1}| = m$, and $W \subseteq T \times M_h \times M_{s,1}$ be an arbitrary instance of problem $P$. Let the elements of these sets be denoted by $T = \{T_1, T_2, \ldots, T_m\}$, $M_h = \{M_{h,1}, M_{h,2}, \ldots, M_{h,m}\}$, $M_{s,1} = \{M_{s,11}, M_{s,12}, \ldots, M_{s,1m}\}$, and $W = \{W_1, W_2, \ldots, W_q\}$, where $|W| = q$. We should construct a set $G$ and a size $s(a) \in Z^+$ for each $a \in G$, such that $G$ contains a subset $G_1$ satisfying

$$\sum_{a \in G_1} s(a) = \sum_{a \in G - G_1} s(a). \tag{13}$$

The set $G$ will contain a total of $q + 2$ elements and will be constructed in two steps.

The first $q$ elements of $G$ are $\{a_k : 1 \le k \le q\}$, where the element $a_k$ is associated with the group $W_k \in W$. The size $s(a_k)$ of $a_k$ will be specified by giving its binary representation, in terms of a string of 0's and 1's divided into $3m$ "zones" of $p = [\log_2(q+1)]$ bits each.

Then each $s(a_k)$ can be expressed in binary with no more than $3pm$ bits; it is clear that $s(a_k)$ can be constructed from the given problem $P$ instance in polynomial time; see Garey and Johnson [21].

If we sum up all elements in any zone, the total can never exceed $q = 2^p - 1$. Therefore, in adding up $\sum_{a \in G_1} s(a)$ for any subset $G_1 \in \{a_k : 1 \le k \le q\}$, there will never be any "carries" from one zone to the next. If we define $B = \sum_{s=0}^{3m-1} 2^{ps}$, then any subset $G_1 \in \{a_k : 1 \le k \le q\}$ will satisfy

$$\sum_{a \in G_1} s(a) = B. \tag{14}$$

The last two elements are denoted by $b_1$ and $b_2$; that is,

$$s(b_1) = 2\sum_{k=1}^{q} s(a_k) - B,$$
$$s(b_2) = \sum_{k=1}^{q} s(a_k) + B. \tag{15}$$

Now suppose we have a subset $G_1 \in G$ such that

$$\sum_{a \in G_1} s(a) = \sum_{a \in G - G_1} s(a). \tag{16}$$

Then both of these sums must be equal to $2\sum_{k=1}^{q} s(a_k)$, and one of the two sets, $G_1$ or $G - G_1$, contains $b_1$ but not $b_2$. It follows that the remaining elements of that set form a subset of $\{a_k : 1 \le k \le q\}$ whose sizes sum to $B$. Therefore the problem $P$ can be transformed to partition, and Proposition 2 is proved. □

*4.2. Solution Procedure.* Since the coordination scheduling problem of multiple suppliers and multiple manufacturers based on the Supply-Hub is NP-complete, the solution may be very complex. Therefore, the auto-adapted differential evaluation algorithm will be proposed to solve this problem by this paper. The differential evolution algorithm put forward by Rainer Storn and Kenneth Price in 1997 is for meta-heuristic global optimization based on population evolutionary and the real coding, which is originally used to solve the Chebyshev polynomials. As to more complex global optimization problems of continuous space, such as non-linear and non-differentiable problems even without function expression, the differential evaluation algorithm has a better global optimization ability and higher convergence performance with simple operation, less controlling parameters, and better robustness, compared to genetic algorithms, particle swarm optimization, simulated annealing, tabu search, and so forth.

The evolution process of differential evaluation algorithm is similar to genetic algorithms, including population initialization, variation, hybridization, and selection. But the main differences between these two algorithms are that the process of variation is before hybridization for differential evolution algorithm, and evaluation of population depends on comparisons with testing chromosome and target chromosome. As a result, the solution procedure of coordination scheduling problem can be proposed as follows.

*4.2.1. Population Initialization.* Let $g$ stands for the generation of population $P^g$, and the scale of population is NP; that is, $P^g = \{x_{i^*}^g\}$, where $i^* = 1, 2, \ldots, \text{NP}$. $x_{i^*}^g$ is a feasible solution of the population $P^g$, which is composed of a vector of $D$ variables, that is $x_{i^*}^g = (x_{i^*1}^g, x_{i^*2}^g, \ldots, x_{i^*D}^g)$.

As for our scheduling problem, $D$ is the number of decision variables. Let $x_{i^*}^g = (x_{i^*1}^g, x_{i^*2}^g, \ldots, x_{i^*D}^g) = (M_{h,1}, M_{h,2}, \ldots, M_{h,m}; M_{s,11}, M_{s,21}, \ldots, M_{s,n1}; M_{s,12}, M_{s,22}, \ldots, M_{s,n2}; \ldots; M_{s,1m}, M_{s,2m}, \ldots, M_{s,nm})$.

Initialize the population, set $g = 0$, and $x_{i^*j^*}^{g=0} = l_{j^*} + \text{rand}_{j^*} \cdot (h_{j^*} - l_{j^*})$.

Where $\text{rand}_{j^*}$ is a real number generated by uniform random distribution in $[0, 1)$, $h_{j^*}$ and $l_{j^*}$ are the upper and lower boundaries of individual variables, which are randomly distrusted real numbers.

*4.2.2. Variations.* The interim of individuals, $v_{i^*}^{g+1} = (v_{i^*1}^{g+1}, v_{i^*2}^{g+1}, \ldots, v_{i^*D}^{g+1})$, should be generated after any individual $x_{i^*}^g$ is determined in population $P^g$, where the number of $x_{i^*}^g$

is $r\,(3 \le r \le \text{NP})$. Let individual set $\Omega = \{\xi_1, \xi_2, \ldots, \xi_r\}$ and after variation the interim individuals $v_{i^*}^{g+1}$ are

$$v_{i^*}^{g+1} = \xi_1 + F \cdot \left[ \left(\xi_2 - \xi_3\right) + \left(\xi_3 - \xi_4\right) + \cdots + \left(\xi_{r-1} - \xi_r\right) \right], \tag{17}$$

where $F$ is a differential scale factor. As for our scheduling problem, the interim individuals $v_{i^*}^{g+1}$ should be rounded to the nearest integer since decision variables $M_{s,ij}$ and $M_{h,j}$ must be positive integers, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$.

*4.2.3. Hybridization.* The interim individuals $v_{i^*}^{g+1}$ should be crossed with current individuals $x_{i^*}^{g}$ in probability CR, where $\text{CR} \in [0, 1]$. The proper individuals can be generated after hybridization. Set $U_{i^*}^{g+1} = (u_{i^*1}^{g+1}, u_{i^*2}^{g+1}, \ldots, u_{i^*D}^{g+1})$. $u_{i^*1}^{g+1}, u_{i^*2}^{g+1}, \ldots, u_{i^*D}^{g+1}$ is a feasible solution of decision variables. Consider

$$u_{i^*j^*}^{g+1} = \begin{cases} v_{i^*j^*}^{g+1} & \text{rand}_{i^*j^*} \le \text{CR or } j = \text{rand}\,(i) \\ x_{i^*j^*}^{g} & \text{else} \end{cases} \tag{18}$$
$$\left(i^* = 1, 2, \ldots, \text{NP};\ j^* = 1, 2, \ , D\right),$$

where CR is the cross rate. The larger the CR is, the more the $u_{i^*j^*}^{g+1}$ can be influenced by $v_{i^*j^*}^{g+1}$, which leads the algorithm to faster convergence with local optimization. In order to increase the performance of differential evolution algorithm, the auto-adapted cross rate was proposed. Let $\text{CR}\,(G_{t=0}) = \text{CR}_{\max}$. When the differential evolution algorithm in the fixed loop of evaluation does not improve significantly, CR can be automatically adapted according to

$$\text{CR}\left(G_{t+1}\right) = \begin{cases} 0.95\text{CR}\left(G_t\right) & \text{if } 0.95\text{CR}\left(G_t\right) \ge \text{CR}_{\min} \\ \text{CR}_{\min} & \text{else,} \end{cases} \tag{19}$$

where $\text{CR}_{\max}$ and $\text{CR}_{\min}$ are the maximum and minimum crossover probabilities, respectively. $G$ is the total evaluation number. $G_{t+1}$ stands for evaluation value in cycle $t + 1$. The auto-adapted change of CR can improve performance of the whole algorithm and enhance the ability of global optimization algorithms.

*4.2.4. Selection.* The fitness of candidate individual $U_{i^*}^{g+1}$ should be evaluated after hybridization. The candidate individual $U_{i^*}^{g+1}$ can be determined whether it replaces the current individuals $x_{i^*}^{g}$ or not according to

$$x_{i^*}^{g+1} = \begin{cases} U_{i^*}^{g+1} & \text{if } C_{\text{csc}}\left(T, U_{i^*}^{g+1}\right) \le C_{\text{csc}}\left(T, x_{i^*}^{g}\right) \\ x_{i^*}^{g} & \text{else,} \end{cases} \tag{20}$$

where $C_{\text{csc}}(\cdot)$ is the fitness function, which corresponds to the total cost of (11), and $T = \{T_1, T_2, \ldots, T_m\}$, where $T_j^*$ $(j = 1, 2, \ldots, m)$ can be calculated from (12). The process should be repeated and the best solution should be output corresponding to $x_{i^*}^{g+1}$ and $T$.

Table 1: Stability analysis of the auto-adapted DE algorithm.

| $m$ | $n$ | Best value | Worst value | Mean | Standard deviation |
|---|---|---|---|---|---|
| 9 | 10 | 38408 | 38633 | 38504 | 42 |
| 7 | 10 | 30160 | 30382 | 30304 | 38 |
| 5 | 10 | 21618 | 21783 | 21706 | 37 |
| 3 | 10 | 12786 | 12937 | 12855 | 31 |
| 3 | 8 | 10524 | 10666 | 10597 | 27 |
| 3 | 6 | 8267 | 8390 | 8311 | 26 |
| 3 | 4 | 5206 | 5418 | 5342 | 31 |

## 5. Numerical Analysis

*5.1. Parameters Setting.* Numerical experiments are conducted to examine the computational effectiveness and efficiency of the proposed auto-adapted differential evaluation algorithm by comparing it with the method of decentralized decision. The parameters of the auto-adapted DE algorithm are as follows: $D = m + m * n$, $\text{NP} = D$, $F_{\min} = 0.2$, $F_{\max} = 0.6$, $\text{CR}_{\min} = 0.2$, $\text{CR}_{\max} = 0.8$, and the maximum number of iterations *GenM* is set at 500 when $m = 10$ and $n = 9, 7, 5$; *GenM* is set at 400 when $n = 3$ and $m = 10, 8$; *GenM* is set at 300 when $n = 3$ and $m = 6, 4$. The detailed settings for each test problem are as follows.

$A_{s,i}$ is selected from uniform distribution $U\,[200, 300]$.

$h_{s,i}$ is selected from uniform distribution $U\,[1, 3]$.

$F_{s,ij} = 30$ and $V_{s,ij}$ is selected from uniform distribution $U\,[10, 20]$.

$A_h = 50$, $h_{h,i} = \alpha_1 \cdot h_{s,i}$, $\alpha_1 = 0.8$.

$F_{h,j} = 5$ and $V_{h,j}$ is selected from uniform distribution $U\,[1, 6]$.

If $w_{ij} = 1$, $d_{ij}$ is selected from uniform distribution $U\,[10, 20]$; otherwise, $d_{ij} = 0$.

$h_{m,j} = \alpha_2 \cdot (\max h_{s,i})$, $\alpha_2 = 3$.

$A_m$ is selected from uniform distribution $U\,[20, 30]$.

*5.2. Comparative Evaluations.* Figures 2, 3, and 4 show the evolution of best solution under 3 different cases, respectively, and we run the proposed auto-adapted DE algorithm under every case for 100 times and calculate its best solutions, worst solutions, means, and standard deviations; the result is shown in Table 1. The results of the auto-adapted DE algorithm and those of the method of decentralized decision are shown in Tables 2 and 3. In Table 2, we assume $n = 10$ and $m = 9, 7, 5, 3$. In Table 3, we assume $m = 3$ and $n = 10, 8, 6, 4$. In both the two tables, $\text{C}_{s,i}$ denotes the cost of supplier $i$; $\text{C}_h$ denotes the cost of Supply-Hub; $\text{C}_{m,j}$ denotes the cost of manufacturer $j$; and $\text{C}_{\text{sc}}$ denotes the cost of supply chain. Table 4 shows the difference of every cost item in the context of joint decision and decentralized decision when $n = 10$. Table 5 shows the difference of every cost item in context of joint decision and decentralized decision when $m = 3$.

From Figures 2, 3, and 4, it can be seen that the auto-adapted DE algorithm is convergent under these 3 cases; in

Table 2: The result of joint decision and decentralized decision when $n = 10$.

| $m$ | Joint decision | | | | | | | Decentralized decision | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M_{h,j}$ | $M_{s,ij}$ | $T_j$ | $C_{s,i}$ | $C_h$ | $C_{m,j}$ | $C_{\text{sc}}$ | $M_{h,j}$ | $M_{s,ij}$ | $T_j$ | $C_{s,i}$ | $C_h$ | $C_{m,j}$ | $C_{\text{sc}}$ |
| 9 | 6 | 4 4 4 2 6 3 4 0 7 4 | 0.24 | 3021 | 6177 | 243 | 37968 | 3 | 7 6 7 5 6 5 5 0 9 6 | 0.23 | 3334 | 5801 | 243 | 40147 |
| | 5 | 3 4 5 3 3 3 5 4 0 3 | 0.28 | 2405 | | 254 | | 3 | 8 6 7 6 7 5 6 8 0 5 | 0.21 | 2647 | | 245 | |
| | 7 | 2 4 0 2 5 3 3 4 5 3 | 0.23 | 2921 | | 266 | | 3 | 6 7 0 5 7 5 6 9 8 5 | 0.21 | 3151 | | 266 | |
| | 6 | 2 0 3 4 7 2 3 4 4 2 | 0.28 | 3858 | | 264 | | 3 | 6 0 6 7 7 5 5 7 9 5 | 0.22 | 4184 | | 258 | |
| | 7 | 4 0 5 3 3 3 3 5 3 2 | 0.23 | 2803 | | 221 | | 4 | 5 0 5 4 6 5 6 7 7 4 | 0.2 | 2939 | | 219 | |
| | 6 | 4 3 4 3 6 3 4 5 4 0 | 0.24 | 2670 | | 226 | | 3 | 6 7 7 6 9 6 5 8 9 0 | 0.21 | 3012 | | 224 | |
| | 6 | 3 3 3 3 0 3 4 5 3 3 | 0.26 | 3273.5 | | 256 | | 3 | 6 5 6 6 0 5 6 7 8 6 | 0.23 | 3545 | | 253 | |
| | 6 | 3 3 3 3 5 2 4 3 6 0 | 0.26 | 3397 | | 241 | | 3 | 7 6 7 6 7 5 6 8 10 0 | 0.21 | 3638 | | 234 | |
| | 6 | 3 3 5 5 0 3 3 4 4 4 | 0.25 | 2800 | | 206 | | 3 | 7 7 7 6 0 6 6 9 10 6 | 0.21 | 3032 | | 203 | |
| | | | | 2465 | | | | | | | 2718 | | | |
| 7 | 8 | 3 3 3 2 3 2 3 0 4 3 | 0.24 | 2392 | 5157 | 243 | 29820 | 3 | 7 6 7 5 6 5 5 0 9 6 | 0.23 | 2638 | 4779 | 243 | 31680 |
| | 5 | 3 3 3 2 4 2 3 4 0 3 | 0.33 | 1696.5 | | 269 | | 3 | 8 6 7 6 7 5 6 8 0 5 | 0.21 | 1895 | | 245 | |
| | 7 | 3 3 0 2 4 2 3 5 5 3 | 0.23 | 2139 | | 267 | | 3 | 6 7 0 5 7 5 6 9 8 5 | 0.21 | 2355 | | 266 | |
| | 6 | 2 0 3 3 5 3 3 4 5 3 | 0.27 | 2962 | | 262 | | 3 | 6 0 6 7 7 5 5 7 9 5 | 0.22 | 3280 | | 258 | |
| | 6 | 3 0 3 2 3 3 4 4 5 3 | 0.28 | 2333 | | 230 | | 4 | 5 0 5 4 6 5 6 7 7 4 | 0.2 | 2517 | | 219 | |
| | 6 | 4 3 4 3 4 4 2 3 5 0 | 0.27 | 2007 | | 230 | | 3 | 6 7 7 6 9 6 5 8 9 0 | 0.21 | 2303 | | 224 | |
| | 8 | 4 3 3 3 0 2 4 5 4 3 | 0.21 | 2464 | | 255 | | 3 | 6 5 6 6 0 5 6 7 8 6 | 0.23 | 2719 | | 253 | |
| | | | | 2574 | | | | | | | 2756 | | | |
| | | | | 2193 | | | | | | | 2365 | | | |
| | | | | 2147 | | | | | | | 2365 | | | |
| 5 | 7 | 3 2 3 2 2 2 2 0 4 3 | 0.29 | 1600 | 3973 | 250 | 21172 | 3 | 7 6 7 5 6 5 5 0 9 6 | 0.23 | 1836 | 3483 | 243 | 22848 |
| | 8 | 3 2 3 2 3 2 2 3 0 2 | 0.26 | 995.5 | | 250 | | 3 | 8 6 7 6 7 5 6 8 0 5 | 0.21 | 1151 | | 245 | |
| | 8 | 2 3 0 2 3 2 2 3 3 2 | 0.25 | 1360.5 | | 270 | | 3 | 6 7 0 5 7 5 6 9 8 5 | 0.21 | 1544 | | 266 | |
| | 8 | 3 0 2 2 3 2 2 3 3 2 | 0.27 | 2170 | | 262 | | 3 | 6 0 6 7 7 5 5 7 9 5 | 0.22 | 2457 | | 258 | |
| | 10 | 2 0 2 2 2 2 2 3 3 2 | 0.24 | 1992 | | 222 | | 4 | 5 0 5 4 6 5 6 7 7 4 | 0.2 | 2202 | | 219 | |
| | | | | 1429 | | | | | | | 1704 | | | |
| | | | | 1676 | | | | | | | 1937 | | | |
| | | | | 1556 | | | | | | | 1721 | | | |
| | | | | 1412 | | | | | | | 1566 | | | |
| | | | | 1754.5 | | | | | | | 2017 | | | |
| 3 | 8 | 3 2 3 2 3 2 2 0 5 2 | 0.26 | 955 | 2330 | 245 | 12786 | 3 | 7 6 7 5 6 5 5 0 9 6 | 0.23 | 1100.5 | 2039 | 243 | 13814 |
| | 7 | 3 2 3 2 3 2 2 3 0 2 | 0.29 | 996 | | 256 | | 3 | 8 6 7 6 7 5 6 8 0 5 | 0.21 | 1151 | | 245 | |
| | 9 | 2 3 0 2 2 2 2 5 4 2 | 0.22 | 617 | | 266 | | 3 | 6 7 0 5 7 5 6 9 8 5 | 0.21 | 702 | | 266 | |
| | | | | 1370 | | | | | | | 1552 | | | |
| | | | | 1294 | | | | | | | 1377 | | | |
| | | | | 968.5 | | | | | | | 1144 | | | |
| | | | | 1042 | | | | | | | 1204 | | | |
| | | | | 761 | | | | | | | 834 | | | |
| | | | | 713 | | | | | | | 776.5 | | | |
| | | | | 1016 | | | | | | | 1181 | | | |

Table 3: The result of joint decision and decentralized decision when $m = 3$.

| | Joint decision | | | | | | | Decentralized decision | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $M_{h,j}$ | $M_{s,ij}$ | $T_j$ | $C_{s,i}$ | $C_h$ | $C_{m,j}$ | $C_{\mathrm{sc}}$ | $M_{s,ij}$ | $M_{s,j}$ | $T_j$ | $C_{s,i}$ | $C_h$ | $C_{m,j}$ | $C_{\mathrm{sc}}$ |
| 10 | 8<br>7<br>9 | 3 2 3 2 3 2 2 0 5 2<br>3 2 3 2 3 2 2 3 0 2<br>2 3 0 2 2 2 2 5 4 2 | 0.26<br>0.29<br>0.22 | 955<br>996<br>617<br>1370<br>1294<br>968.5<br>1042<br>761<br>713<br>1016 | 2330 | 245<br>256<br>266 | 12786 | 3<br>3<br>3 | 7 6 7 5 6 5 5 0 9 6<br>8 6 7 6 7 5 6 8 0 5<br>6 7 0 5 7 5 6 9 8 5 | 0.23<br>0.21<br>0.21 | 1100.5<br>1151<br>702<br>1552<br>1377<br>1144<br>1204<br>834<br>776.5<br>1181 | 2039 | 243<br>245<br>266 | 13814 |
| 8 | 8<br>8<br>8 | 2 2 2 2 2 2 2 0<br>3 2 2 2 3 2 2 2<br>2 2 0 2 2 2 2 3 | 0.3<br>0.28<br>0.28 | 939.5<br>979<br>600<br>1361<br>1234<br>963<br>1031<br>742 | 1980.5 | 223<br>234<br>236 | 10524 | 3<br>3<br>3 | 6 5 6 4 6 4 5 0<br>7 5 7 6 6 4 6 8<br>5 6 0 4 6 4 5 6 | 0.25<br>0.23<br>0.24 | 1085<br>1134.5<br>693<br>1536<br>1362<br>1127<br>1187<br>824 | 1703.5 | 220<br>229<br>234 | 11335 |
| 6 | 7<br>6<br>7 | 3 2 2 2 2 2<br>3 2 3 2 3 2<br>2 2 0 2 2 1 | 0.3<br>0.3<br>0.4 | 948<br>980<br>610<br>1365<br>1232<br>947 | 1570 | 204.5<br>205<br>205.5 | 8267 | 3<br>3<br>3 | 6 5 6 4 5 4<br>6 5 6 4 6 4<br>5 5 0 4 5 4 | 0.28<br>0.26<br>0.28 | 1066<br>1116<br>682<br>1512<br>1344<br>1105 | 1380 | 202<br>199<br>200 | 8806 |
| 4 | 6<br>6<br>8 | 2 2 2 2<br>2 2 2 2<br>1 2 0 1 | 0.4<br>0.4<br>0.5 | 892<br>966<br>594<br>1304 | 1057 | 162<br>155<br>154 | 5284 | 3<br>3<br>3 | 4 4 5 3<br>5 3 4 4<br>3 4 0 3 | 0.35<br>0.34<br>0.37 | 1030<br>1079<br>661.5<br>1470 | 895 | 159.5<br>152<br>150 | 5597 |

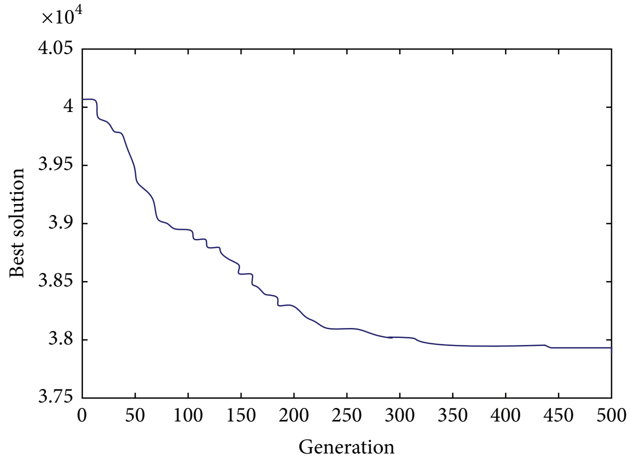


FIGURE 2: The evolution of best solution when $m = 9$, $n = 10$.

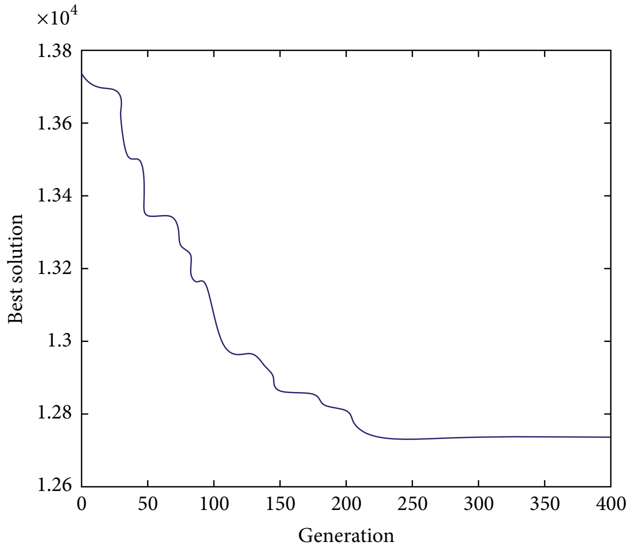


FIGURE 3: The evolution of best solution when $m = 3$, $n = 10$.

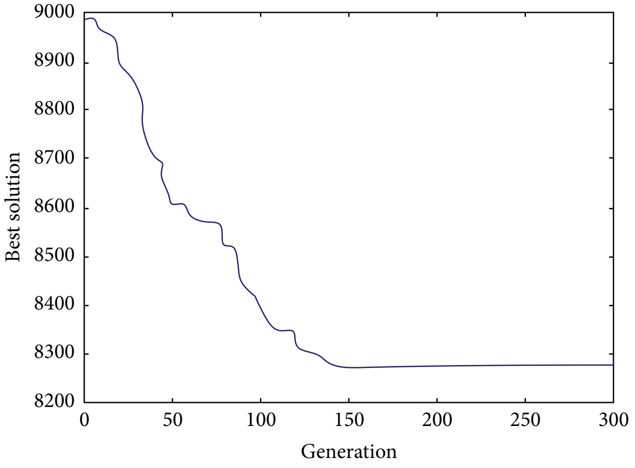


FIGURE 4: The evolution of best solution when $m = 3$, $n = 6$.

TABLE 4: The difference of every cost item when $n = 10$.

| $m$ | 9 | 7 | 5 | 3 |
|---|---|---|---|---|
| $\Delta C_{s1}$ | −9.4% | −9.3% | −12.9% | −13.2% |
| $\Delta C_{s2}$ | −9.1% | −10.4% | −13.5% | −13.5% |
| $\Delta C_{s3}$ | −7.3% | −9.2% | −11.9% | −12.1% |
| $\Delta C_{s4}$ | −7.8% | −9.7% | −11.7% | −11.7% |
| $\Delta C_{s5}$ | −4.6% | −7.3% | −9.5% | −6% |
| $\Delta C_{s6}$ | −11.4% | −12.9% | −16.1% | −15.3% |
| $\Delta C_{s7}$ | −7.6% | −9.4% | −13.5% | −13.5% |
| $\Delta C_{s8}$ | −6.6% | −6.6% | −9.6% | −8.8% |
| $\Delta C_{s9}$ | −7.7% | −7.3% | −9.8% | −8.1% |
| $\Delta C_{s10}$ | −9.3% | −9.2% | −13% | −14% |
| $\Delta C_{m1}$ | 0% | 0% | 2.9% | 0.8% |
| $\Delta C_{m2}$ | 3.7% | 9.8% | 2% | 4.5% |
| $\Delta C_{m3}$ | 0% | 0.4% | 1.5% | 0% |
| $\Delta C_{m4}$ | 2.3% | 1.6% | 1.6% | |
| $\Delta C_{m5}$ | 0.9% | 5% | 1.4% | |
| $\Delta C_{m6}$ | 0.9% | 2.7% | | |
| $\Delta C_{m7}$ | 1.2% | 0.8% | | |
| $\Delta C_{m8}$ | 3% | | | |
| $\Delta C_{m9}$ | 1.5% | | | |
| $\Delta C_{h}$ | 6.5% | 7.9% | 14.1% | 14.3% |
| $\Delta C_{sc}$ | −5.4% | −5.9% | −7.3% | −7.4% |

TABLE 5: The change of every cost item when $m = 3$.

| $n$ | 10 | 8 | 6 | 4 |
|---|---|---|---|---|
| $\Delta C_{m1}$ | 0.8% | 1.4% | 1.2% | 1.3% |
| $\Delta C_{m2}$ | 4.5% | 2.2% | 3% | 2% |
| $\Delta C_{m3}$ | 0% | 0.9% | 2.8% | 2.7% |
| $\Delta C_{s1}$ | −13.2% | −13.4% | −11.1% | −13.4% |
| $\Delta C_{s2}$ | −13.5% | −13.7% | −12.2% | −10.5% |
| $\Delta C_{s3}$ | −12.1% | −13.4% | −10.6% | −10.3% |
| $\Delta C_{s4}$ | −11.7% | −11.4% | −9.7% | −11.3% |
| $\Delta C_{s5}$ | −6% | −9.4% | −8.3% | |
| $\Delta C_{s6}$ | −15.3% | −14.6% | −14.3% | |
| $\Delta C_{s7}$ | −13.5% | −13.1% | | |
| $\Delta C_{s8}$ | −8.8% | −10% | | |
| $\Delta C_{s9}$ | −8.1% | | | |
| $\Delta C_{s10}$ | −14% | | | |
| $\Delta C_{h}$ | 14.3% | 16.3% | 13.8% | 18.1% |
| $\Delta C_{sc}$ | −7.4% | −7.2% | −6.1% | −5.6% |

fact, the algorithm is convergent under all these 7 cases in our numerical experiment; we only show the 3 figures due to the limited space. From Table 1 we can see that even under the case $m = 9$ and $n = 10$, the standard deviation is relatively small, so we can conclude that the auto-adapted DE algorithm is stable.

From Tables 2, 3, 4, and 5, we can obtain some conclusions as follows.

(1) When suppliers, the Supply-Hub, and manufacturers make decisions as a whole, the total cost of supply

chain can be reduced compared to the corresponding cost when they make decisions decentralized. Tables 4 and 5 reveal that the total cost of supply chain can be reduced by 5.4% at least, 7.4% at most.

(2) When suppliers, the Supply-Hub, and manufacturers make decisions centralized, every supplier's cost decreases, but the Supply-Hub's cost and every manufacturer's cost increases, and the decreased cost is more than the increased one, so the total cost of supply chain can be reduced. We can see that the Supply-Hub's cost increases greatly in context of centralized decision-making from Tables 4 and 5, so the operator of the Supply-Hub may be not willing to make decisions centralized. In fact, suppliers always sell their products to the manufacturer on consignment under Supply-Hub mode. The inventory holding cost is paid by suppliers when their products are stored in the Supply-Hub, as every supplier's cost decreases greatly on the condition of centralized decision-making, so they are willing to pay the increased inventory holding cost.

(3) The Supply-Hub's distribution interval and every supplier's distribution interval increase under centralized decision-making compared to the results obtained in the case of decentralized decision-making, but for supplier's order interval, some increase and others decrease. From Tables 2 and 3, it can be seen that every supplier's distribution interval and $M_{h,j}$ increase in the context of centralized decision-making, so the Supply-Hub's distribution interval for every manufacturer also increases under this case.

(4) From Table 2, it can be seen that in case of decentralized decision-making, all the suppliers' and Supply-Hub's decisions remain the same as the number of manufacturer increases, but under centralized decision-making, their decisions change as the number of manufacturer increases. This is because in the context of centralized decision-making, every decision maker considers the influence of his decision on others, and they optimize the whole supply chain collaboratively. Therefore, as the number of manufacturer increases, all the suppliers and the Supply-Hub change their optimal decisions.

## 6. Conclusions

This paper examines the collaborative scheduling model for the Supply-Hub consists of multiple suppliers and multiple manufacturers. We describe the basic operational process of the Supply-Hub and formulate the basic decision models. Given two different scenarios of decentralized system and collaborative system, we first consider the case that the Supply-Hub, the suppliers, and the manufacturers operate separately in their delivery quantities, production quantities, and order quantities. We next consider the collaborative mechanism, in which the Supply-Hub makes the entire decisions for all the suppliers and manufacturers. Furthermore, we offer the complexity analysis for the collaborative scheduling model and it turns to be proved NP-complete. Consequently, we propose an auto-adapted differential evolution algorithm. The numerical analysis illustrates that the performance of collaborative decision is superior to the decentralized decision. All these results demonstrate that the implementation of Supply-Hub can significantly reduce the operation cost in the assembly system, and thus improve the supply chain's overall performance.

## Conflict of Interests

The authors declare no conflict of interests. We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, "A Collaborative Scheduling Model for the Supply-Hub with Multiple Suppliers and Multiple Manufacturers".

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 71102174, 71372019, and 71231007), Specialized Research Fund for Doctoral Program of Higher Education of China (no. 20111101120019), Beijing Philosophy and Social Science Foundation of China (no. 11JGC106), Beijing Higher Education Young Elite Teacher Project (no. YETP1173) and China Postdoctoral Science Foundation (no. 2013M542066).

## References

[1] E. Barnes, J. Dai, S. Deng et al., *On the Strategy of Supply-Hubs for Cost Reduction and Responsiveness*, National University of Singapore, Singapore, 2000.

[2] Y. Wang and S. Ma, "A study on supply-hub mode under the supply chain environment," *Management Review*, vol. 17, no. 2, pp. 33–36, 2005 (Chinese).

[3] J. Shah and M. Goh, "Setting operating policies for supply hubs," *International Journal of Production Economics*, vol. 100, no. 2, pp. 239–252, 2006.

[4] J. Hahm and C. A. Yano, "The economic lot and delivery scheduling problem: the single item case," *International Journal of Production Economics*, vol. 28, no. 2, pp. 235–252, 1992.

[5] J. Hahm and C. A. Yano, "Economic lot and delivery scheduling problem: the common cycle case," *IIE Transactions*, vol. 27, no. 2, pp. 113–125, 1995.

[6] J. Hahm and C. A. Yano, "Economic lot and delivery scheduling problem: models for nested schedules," *IIE Transactions*, vol. 27, no. 2, pp. 126–139, 1995.

[7] M. Khouja, "The economic lot and delivery scheduling problem: common cycle, rework, and variable production rate," *IIE Transactions*, vol. 32, no. 8, pp. 715–725, 2000.

[8] J. Clausen and S. Ju, "A hybrid algorithm for solving the economic lot and delivery scheduling problem in the common cycle case," *European Journal of Operational Research*, vol. 175, no. 2, pp. 1141–1150, 2006.

[9] F. E. Vergara, M. Khouja, and Z. Michalewicz, "An evolutionary algorithm for optimizing material flow in supply chains," *Computers and Industrial Engineering*, vol. 43, no. 3, pp. 407–421, 2002.

[10] M. Khouja, "Synchronization in supply chains: implications for design and management," *Journal of the Operational Research Society*, vol. 54, no. 9, pp. 984–994, 2003.

[11] G. Pundoor, *Integrated Production-Distribution Scheduling in Supply Chains*, University of Maryland, College Park, Md, USA, 2005.

[12] S. A. Torabi, S. M. T. Fatemi Ghomi, and B. Karimi, "A hybrid genetic algorithm for the finite horizon economic lot and delivery scheduling in supply chains," *European Journal of Operational Research*, vol. 173, no. 1, pp. 173–189, 2006.

[13] D. Naso, M. Surico, B. Turchiano, and U. Kaymak, "Genetic algorithms for supply-chain scheduling: a case study in the distribution of ready-mixed concrete," *European Journal of Operational Research*, vol. 177, no. 3, pp. 2069–2099, 2007.

[14] S. Ma and F. Gong, "Collaborative decision of distribution lot-sizing among suppliers based on supply-hub," *Industrial Engineering and Management*, vol. 14, no. 2, pp. 1–9, 2009 (Chinese).

[15] C. Lin and S. Chen, "An integral constrained generalized hub-and-spoke network design problem," *Transportation Research E*, vol. 44, no. 6, pp. 986–1003, 2008.

[16] C. Lin, "The integrated secondary route network design model in the hierarchical hub-and-spoke network for dual express services," *International Journal of Production Economics*, vol. 123, no. 1, pp. 20–30, 2010.

[17] V. T. Charles, Y. P. L. Gilbert, J. C. T. Amy, C. S. Liu, and W. T. Lee, "Deriving industrial logistics hub reference models for manufacturing based economies," *Expert System with Applications*, vol. 38, pp. 1223–1232, 2011.

[18] J. Li, S. Ma, P. Guo, and C. Liu, "Supply chain design model based on BOM-Supply Hub," *Computer Integrated Manufacturing Systems*, vol. 15, no. 7, pp. 1299–1306, 2009 (Chinese).

[19] H. Gui and S. Ma, "A study on the multi-source replenishment model and coordination lot size decision-making based on Supply-Hub," *Chinese Journal of Management Science*, vol. 18, no. 1, pp. 78–82, 2010 (Chinese).

[20] G. Li, S. Ma, F. Gong, and Z. Wang, "Research reviews and future prospective of collaborative operation in supply logistics based on Supply-Hub," *Journal of Mechanical Engineering*, vol. 47, no. 20, pp. 23–33, 2011 (Chinese).

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, A Series of Books in the Mathematical Sciences, W.H. Freeman and Company, New York, NY, USA, 1979.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 870892, 12 pages
http://dx.doi.org/10.1155/2014/870892

*Research Article*

# Comprehensive Optimization of Emergency Evacuation Route and Departure Time under Traffic Control

**Guo Li,[1] Ying Zhou,[1] and Mengqi Liu[2]**

[1] *School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China*
[2] *Business School, Hunan University, Changsha 410082, China*

Correspondence should be addressed to Mengqi Liu; 1069679071@qq.com

Received 2 November 2013; Accepted 4 December 2013; Published 13 April 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

With the frequent occurrence of major emergencies, emergency management gets high attention from all around the world. This paper investigates the comprehensive optimization of major emergency evacuation route and departure time, in which case the evacuation propagation mechanism is considered under traffic control. Given the practical assumptions, we first establish a comprehensive optimization model based on the simulation of evacuation route and departure time. Furthermore, we explore the reasonable description method of evacuation traffic flow propagation under traffic control, including the establishment of traffic flow propagation model and the design of the simulation mudule that can simulate the evacuation traffic flow. Finally, we propose a heuristic algorithm for the optimization of this comprehensive model. In case analysis, we take some areas in Beijing as the evaluation sources to verify the reliability of our model. A series of constructive suggestions for Beijing's emergency evacuation are proposed, which can be applied to the actual situation under traffic control.

## 1. Introduction

In recent years we have witnessed a significant increase in the incidence of major natural disasters. For example, disastrous emergencies such as SARS, Indonesian tsunami, and Japan earthquake occur frequently all around the world. On a global scale, the losses caused by disasters such as super tornadoes in the southern United States, floods in Queensland, and the earthquakes that hit Australia and New Zealand reached the level which only happened once in ten years or even decades. Also, the Japanese earthquake and the following nuclear leak as a whole became the second largest catastrophic event in the history (measured by the insurance loss). The total economic losses caused by disasters in the first half of 2011 were close to $278 billion, which were second only to the losses in 2005 (the most serious year in the history). For China, many major natural disasters such as severe snow and ice in northern China, the 5.8 magnitude earthquake in Yunnan, Yingjiang, the rainstorm and flood in southern China, and coastal typhoon occurred one after another, and all of that mentioned above had great impacts on the development of economy and society, people's life, and property safety. According to the statistics, in 2011, due to natural disasters, 430 million people were affected, 1126 people were killed (including the missing 112 people), 9.394 million people were evacuated, and the direct economic losses hit 309.64 billion RMB (excluding the data in Hong Kong, Macao, and Taiwan).

With the further acceleration of urbanization, population becomes more concentrated. Thus, it is important to figure out how to respond to the emergencies timely and effectively. Emergency evacuation, as a main means of rapid evacuation and reducing losses to the minimum in times of crisis, has been on the front burner in the emergency management.

According to the scale, evacuation issues can be divided into small scale evacuation and long-distance regional evacuation. Small scale evacuation generally refers to the evacuation aiming at swift and violent emergencies that only affect small space, such as explosion within a finite range, house collapse, and fire breaking out in shopping malls. Usually, evacuation of this kind is mainly dealt with walking evacuation artificially, not requiring means of transport. While the long-distance regional evacuation generally needs vehicles for transport, and it is often commanded by the government or related departments with mandatory measures. Therefore, the evacuation problems discussed in this paper all belong to long-distance regional and mandatory evacuation.

The rest of our paper is organized as follows. Section 2 reviews the related literature. Section 3 establishes a comprehensive optimization model based on the simulation of evacuation route and departure time. Section 4 develops the reasonable description method of evacuation traffic flow propagation under traffic control. Section 5 provides a heuristic algorithm for the optimization of this comprehensive model. Case analysis and tests are reported in Section 6. The concluding remarks are given in Section 7.

## 2. Literature Review

Our paper belongs to the vast literature that investigates the optimization of emergency evacuation route and departure time. In the prior literature, many scholars have studied this issue from different perspectives, for example, emergency evacuation model [1–8], evacuation simulation [9–13], optimal route finding [14, 15], and the algorithm [16–18]. Among them, Antoine and Erol [1] propose an approach that offers fast estimates based on graph models and probability models. They show that graph models can offer insight into the critical areas in an emergency evacuation and that analytical models based on queuing theory can provide useful estimates for the optimization of evacuation time and route. Chen and Xiao [4] establish an optimal objective based on the shortest emergency time and acquire the optimal solution using the Pontryagin minimum principle. The evacuation route construction algorithm and traffic flow assignment algorithm in each junction are employed to deliver the traffic flow in the evacuation area to a safe region rapidly and safely. The idea of feedback is introduced in the execution using real-time information to adjust and update the evacuation plan. Hasby and Khodra [14] discuss Twitter-based traffic information extraction and its usage as heuristic in optimal route finding. Their system is divided into two modules, extraction information and route finding, and displays a map with marked route based on traffic information extracted from Twitter.

Lu and Betsy [17] present a heuristic algorithm, namely, capacity constrained route planner (CCRP), which models capacity as a time series and uses a capacity constrained routing approach to incorporate route capacity constraints. The CCRP algorithm produces high quality solutions and significantly reduces the computational cost compared to linear programming approach. Papinigis et al. [19] present a calculation method for people evacuation based on the physical characteristics of people stream (density, intensity, and movement speed). The time required to evacuate people from the building is determined in the numerical illustration of the method application. Furthermore, they make the comparison between simple calculation method and modeling with FDS + Evac software.

In addition, on the evacuation problem of nuclear power plant surrounding areas, Dunn and Newton [20] treat evacuation traffic flow and evacuation time as quantitative, explore how to evacuate people as far as possible within the prescribed time, and consider the evacuation as maximum-flow problem to find the optimal evacuation route. On the evacuation problem of earthquake, Yamada [21] describes route optimization of neighborhoods getting to their respective shelters as shortest route problem and the minimum cost flow problem. By finding the shortest route and minimum cost flow, evacuation planning is obtained. Campos et al. [22] firstly analyze how to reduce the conflict between the evacuation traffic flow and increase the traffic capacity of evacuation route, then treat the ratio of the traffic capacity of route and travel time as a measurable index of traffic performance, finally put forward a heuristic algorithm to find the best route between OD points. Satoko et al. [23] describe the optimization of evacuation route and departure time as the fastest flow problem under the premise that the traffic starting from the same point can only depart for a designated ending point, and they provide an algorithm to find the fastest flow in the tree network.

Our paper differs from the above literature in the following aspects. First, traffic flow propagation in emergency evaluation should be considered as it affects the evaluation measures. Besides, traffic control and regulations in turn will have great effects on traffic flow propagation. As a result, our paper investigates the optimization of emergency evacuation route and departure time considering the interaction between traffic control and traffic flow propagation. Moreover, we not only establish a simulation model but also design a similar heuristic algorithm. At last, we choose some areas in Beijing to verify the reliability of the model.

## 3. Comprehensive Optimization Model

*3.1. Assumptions and Parameter Definitions.* A comprehensive optimization model is established based on the simulation of evacuation route and departure time and its premises and assumptions are as follows.

(1) The traffic volume (or the number of vehicles) at evacuation source that needs to be evacuated to the ending point is known.

(2) The traffic volume is 0 at the initial time in the road network, regardless of the traffic that has already existed in the network.

(3) There exist only evacuation traffic flow and rescue traffic flow in the network during the evacuation period, and evacuation traffic flow is effectively separate from the rescue traffic flow under traffic control.

Nodes include the intersection points, evacuation sources, and ending points. Arcs mean the distance between adjacent intersections or the distance between the intersection and evacuation source or ending point. By removing the arcs with one-way and prohibition sections and changing the number of lanes and traffic capacity of the arcs with the sections that have reverse traffic, the evacuation network can be abstracted into a directed network $G = \{V, A\}$, where $V$ is the set of nodes and $A$ is the set of arcs. Assume that the evacuation happens in $[0,T]$, which means that the starting time of the evacuation is 0 and the ending time is $T$. Divide $[0,T]$ into $K$ parts, each of which is $\Delta$, so $T = K\Delta$. Symbols and variables in this paper are defined as follows.

$M$: set of evacuation source nodes in the road network.

$N$: set of evacuation ending nodes in the road network.

$m$: evacuation source node, where $m \in M$.

$n$: evacuation ending node, where $n \in N$.

$a$: section in the evacuation network, where $a \in A$.

$p$: intersection in the evacuation network, where $p \in V$.

$R_{mn}$: set of routes between the evacuation source $m$ and the ending node $n$.

$l$: serial-number of the period in which vehicles depart from the evacuation source, where $l = 1, 2, \ldots, K$.

$l_s$, $l_e$: starting (ending) point of period $l$, $l_s = (l-1)\Delta$, $l_e = l_s + \Delta$.

$k$: serial-number of the period in which vehicles run in the road network, $k = 1, 2, \ldots, K$.

$k_s$, $k_e$: starting (ending) point of period $k$, $k_s = (k-1)\Delta$, $k_e = k_s + \Delta$.

$D_{mn}$: total evacuation demand (or the number of vehicles) between $m$ and $n$.

$F^l_{mnr}$: number of vehicles that depart from $m$ to $n$ along route $r$ in period $l$, where $r \in R_{mn}$.

$F$: set of $F^l_{mnr}$, for all $m, n, r, l$.

$p^k_a$: number of vehicles running on section $a$ at the starting point of period $k$.

$x^k_a$, $y^k_a$: number of vehicles that enter (leave) section $a$ in period $k$.

$x^{lka}_{mnr}$, $y^{lka}_{mnr}$: number of vehicles that enter (leave) section $a$ in period $k$ among those that depart from $m$ to $n$ along route $r$ in period $l$.

$t^l_{mnr}(F)$: average running time spent by vehicles that depart from $m$ to $n$ along route $r$ in period $l$ under the traffic condition determined by $F$.

$A(p)$, $B(p)$: set of sections, and of which the starting (ending) nodes are $p$.

$\delta^{lka}_{mnr}$: value is 0 or 1. If vehicles that depart from $m$ to $n$ along route $r$ in period $l$ enter section $a$ in period $k$, then 1 is taken; otherwise 0 is taken.

*3.2. Model Formulation.* The objective of an evacuation organization planning is to get the shortest total evacuation time with reasonable arrangement for departure time and evacuation route, which can be described as

$$\operatorname{Min} T. \tag{1}$$

Though formula (1) is intuitive and simple, it is unable to establish contact with decision variables. Therefore, the following formula can be expanded from formula (1) as the objective function:

$$\operatorname{Min} Z(F) = \sum_{m\in M}\sum_{n\in N}\sum_{r\in R_{mn}}\sum_{l=1}^{K} F^l_{mnr}\left[(l_s + \alpha\Delta) + t^l_{mnr}(F)\right], \tag{2}$$

where $l_s + \alpha\Delta$ means the average waiting time spent by vehicles that depart in period $l$ at evacuation source, and $0 \le \alpha \le 1$.

Constraints of the model are as follows:

$$\sum_{r\in R_{mn}}\sum_{1\le l\le K} F^l_{mnr} = D_{mn} \quad \forall m, n, \tag{3}$$

$$p_a(0) = 0 \quad \forall a, \tag{4}$$

$$p^k_a = p^{k-1}_a + x^k_a - y^k_a \quad \forall k, a, \tag{5}$$

$$\sum_{b\in B(p_1)} x^k_b = \sum_{c\in A(p_2)} y^k_c \tag{6}$$

$$\forall k, p_1 \in A - M, p_2 \in A - N,$$

$$x^k_a = \sum_{m\in M}\sum_{n\in N}\sum_{r\in R_{mn}}\sum_{1\le l\le K} x^{lka}_{mnr} \quad \forall k, a, \tag{7}$$

$$y^k_a = \sum_{m\in M}\sum_{n\in N}\sum_{r\in R_{mn}}\sum_{1\le l\le K} y^{lka}_{mnr} \quad \forall k, a, \tag{8}$$

$$\begin{pmatrix} x^{lka}_{mnr} \\ y^{lka}_{mnr} \\ \delta^{lka}_{mnr} \end{pmatrix} = g\left(F^l_{mnr} : \forall m, n, r, l\right) \quad \forall m, n, r, k, l, a, \tag{9}$$

$$t^l_{mnr}(F) = \sum_{a\in A}\sum_{k=1}^{K} \delta^{lka}_{mnr} \cdot \Delta \quad \forall m, n, r, l, \tag{10}$$

where all variables are greater than or equal to 0.

Formula (3) is the conservation constraint of total evacuation demand, which means that the total number of vehicles that depart along all dynamic routes between the evacuation OD is equal to the total evacuation demand. Formula (4) gives the initial condition of the road network, which means there is no vehicle at the initial time. Formula (5) is the state equation of the section, which describes the dynamic changes of the number of vehicles in the section. Formula (6) is the conservation constraint of the node, which means vehicles cannot stop at the node, and in each period the total number of vehicles arriving at a node is equal to the total number of vehicles leaving this node. Formulas (7) and (8) give the calculation method of outflow and inflow of the dynamic section. Formula (9) represents a constraint of traffic flow propagation, in which the function $g()$ describes the relationship between the state variables based on section and the decision variables based on route. The detailed description will be illustrated by simulation method in Section 4. Formula (10) is used to calculate the running time on the route and it also shows the relationship between decision variables and objective function.

## 4. Traffic Flow Propagation Mechanism

*4.1. Model Description and Parameter Definitions.* In this section, we will explore the traffic flow propagation mechanism. In order to reflect the bottleneck effect which is exerted by control measures at the intersection in the process of traffic flow propagation, we need to, respectively, describe the dynamic outflow of each entrance lane. In addition, regardless of the microbehaviors of vehicles in the running process, such as lane changing and overtaking, it can be argued that traffic flow propagation in a section (or in the running process) is subject to the same law in an average sense.

Thus, this paper establishes the comprehensive model that includes both running time function and shunted outflow function to describe the traffic flow propagation in a section and between two sections at the intersection. The model is divided into two parts in theory: the running part and the queuing part. Running time function describes the traffic flow propagation in the running part, and shunted outflow function describes the traffic flow propagation in the queuing part. Shunted outflow function divides the queuing part at the end of the section into multiple queues, and the traffic flow propagation between sections can be described by the dynamic outflow rate of each queue.

We use $G = \{V, A\}$ to stand for the evacuation network and divide the period $[0, T]$ into $M$ intervals. The length of each interval is $\sigma$ ($\sigma < \Delta$), which is small enough to guarantee that the inflow and outflow rates of the section remain unchanged in each interval. In order to distinguish $\sigma$ and $\Delta$, here we assume time $t$ is a small period whose length is $\sigma$. For the sake of simplicity, we convert all time variables to integer times of $\sigma$. Symbols and variables in this model are defined as follows.

$t$: current (discretization) time, $t = 1, 2, \ldots, M$.

$f_{mnr}(t)$: departure (or outflow) rate of route $r$ between $m$ and $n$ at time $t$, where $r \in R_{mn}$.

$xr_{mnr}^{a}(t)$, $yr_{mnr}^{a}(t)$: flow rate while vehicles that depart from $m$ to $n$ along route $r$ are entering (leaving) the running part of section $a$ at time $t$.

$xq_{mnr}^{aa^{+}}(t)$, $yq_{mnr}^{aa^{+}}(t)$: flow rate while vehicles that depart from $m$ to $n$ along route $r$ are entering (leaving) the turning queue from section $a$ to section $a^{+}$ (the downstream section of this turning is section $a^{+}$) at time $t$.

$nr_{mnr}^{a}(t)$: number of vehicles that depart from $m$ to $n$ along route $r$ in the running part of section $a$ at time $t$.

$nq_{mnr}^{aa^{+}}(t)$: number of vehicles that depart from $m$ to $n$ along route $r$ among those that are in the turning queue from section $a$ to section $a^{+}$ at time $t$.

$nr_{a}(t)$: number of vehicles that are in the running part of section $a$ at time $t$.

$nq_{aa^{+}}(t)$: number of vehicles that are in the turning queue from section $a$ to section $a^{+}$ at time $t$.

$nq_{a}(t)$: number of vehicles that are in the queuing part of section $a$ at time $t$, and it is equal to the sum of the number of vehicles in all turning queues at that time.

$xq_{aa^{+}}(t)$, $yq_{aa^{+}}(t)$: inflow rate (outflow rate) of the turning queue from section $a$ to section $a^{+}$ at time $t$.

$xq_{a}(t)$,$yq_{a}(t)$: total inflow (outflow) rate of section $a$, and it is equal to the sum of the inflow (outflow) rates of all turning queues at time $t$.

$tr_{a}(t)$: running time of vehicles that enter section $a$ at time $t$.

$tq_{aa^{+}}(t)$: waiting time (or the delay) of vehicles that enter the turning queue at time $t$ after leaving the running part of section $a$.

$h_{a}$: maximum number of vehicles in section $a$ (associated with the geometric conditions and jam density of the section).

$s_{aa^{+}}$: maximum outflow rate of the turning queue from section $a$ to section $a^{+}$ (determined by the capacity of the entrance lane in this turning direction).

$s_{a}$: maximum inflow rate of section $a$ (determined by the capacity of this section).

$a_{h}$, $a_{t}$: terminal node (beginning node) of section $a$.

*4.2. Running Part.* Traffic flow propagation in the running part is described by running time function, namely, the transformation of basic speed formula:

$$T' = \frac{L}{V}, \tag{11}$$

where $T'$ is the running time, $L$ is the length of the section, and $V$ is the running speed. Due to the existence of intersections, the actual running distance is smaller than the length of the section. But compared to the running distance, the distance at the intersection is small enough to be ignored. Therefore, here, the running distance can be the substitute of the length of the section. The limit of the running speed is the maximum speed, which varies in different sections.

In section $a$, the running time can be calculated:

$$tr_{a}(t) = \frac{L_{a}}{V_{a}}. \tag{12}$$

The aim of the description of traffic flow propagation in the running part is to get its outflow rate based on the inflow rate, running time and number of vehicles in the section, as shown in the following formulas:

$$yr^{a}_{mnr}\left(t + tr_{a}\left(t\right)\right) = \frac{xr^{a}_{mnr}\left(t\right)}{l + tr_{a}\left(t\right) - tr_{a}\left(t-1\right)}, \quad \forall m, n, t, a, \ \forall r \in R_{mn}, \tag{13}$$

$$yr^{a}_{mnr}\left(i\right) = yr^{a}_{mnr}\left(j\right) + \left(i - j\right)\frac{yr^{a}_{mnr}\left(j'\right) - yr^{a}_{mnr}\left(j\right)}{j' - j}, \quad \forall m, n, a, \quad \forall r \in R_{mn}, \tag{14}$$

where $i$ is an integer, $j = t + tr_{a}(t)$, and $j' = t + 1 + tr_{a}(t+1)$.

Formula (14) is the linear approximation of the outflow rates that are not at integer time.

The state of the running part, namely, the number of vehicles, can be updated by the following formulas:

$$nr^{a}_{mnr}\left(t+1\right) = nr^{a}_{mnr}\left(t\right) + xr^{a}_{mnr}\left(t\right) - yr^{a}_{mnr}\left(t\right) \tag{15}$$

$$nr_{a}\left(t\right) = \sum_{m \in M}\sum_{n \in N}\sum_{\substack{r: \in R_{mn} \\ a \in r}} nr^{a}_{mnr}\left(t\right), \quad \forall a, t. \tag{16}$$

*4.3. Queuing Part.* The purpose of the description of traffic flow propagation in the queuing part is mainly to get the actual outflow rate based on the outflow demand and downstream supply. The outflow demand of the queue is determined by the inflow rate, number of vehicles in the queue and biggest outflow rate of the entrance lane at that time, and the downstream supply is determined by the space capacity, maximum inflow rate and total outflow rate at that time.

The inflow rate of each turning queue is equal to the outflow rate in the corresponding turning direction in the running part, as shown in following formulas:

$$xq^{aa^{+}}_{mnr}\left(t\right) = yr^{a}_{mnr}\left(t\right), \quad \forall m, n, r \ \forall r \in R_{mn}, a, a^{+} \in r, \tag{17}$$

$$xq_{aa^{+}}\left(t\right) = \sum_{m \in M}\sum_{n \in N}\sum_{\substack{r \in R_{mn} \\ a, a^{+} \in r}} xq^{aa^{+}}_{mnr}\left(t\right), \quad \forall a, a^{+} : a_{h} = a^{+}_{t}. \tag{18}$$

The outflow rate in the queuing part can be calculated by the following formula based on its own outflow demand and downstream supply:

$$yq_{aa^{+}}\left(t\right) = \min\left[M_{aa^{+}}\left(t\right), N_{aa^{+}}\left(t\right)\right] \quad \forall a, a^{+} : a_{h} = a^{+}_{t}. \tag{19}$$

$N_{aa+}(t)$ and $M_{aa^{+}}(t)$, respectively, represent the outflow demand in the turning queue from section $a$ to section $a^{+}$ and the supply capacity in section $a^{+}$ that can be assigned to section $a$ at time $t$. $N_{aa^{+}}(t)$ can be obtained by formula (20), and $M_{aa^{+}}(t)$ can be calculated by formulas (21) and (22).

Consider the following:

$$N_{aa^{+}}\left(t\right) = \begin{cases} xq_{aa^{+}}\left(t\right), & \text{if } nq_{aa^{+}}\left(t\right) = 0, \quad xq_{aa^{+}}\left(t\right) < s_{aa^{+}} \\ s_{aa^{+}}, & \text{otherwise,} \end{cases} \quad \forall a, a^{+} : a_{h} = a^{+}_{t}, \tag{20}$$

$$M_{aa^{+}}\left(t\right) = \beta_{aa^{+}} \cdot R_{a^{+}}\left(t\right) \quad \forall a, a^{+} : a_{h} = a^{+}_{t}, \tag{21}$$

$$R_{a^{+}}\left(t\right) = \begin{cases} s_{a^{+}}, & \text{if } x_{a^{+}}\left(t\right) < h_{a^{+}} \\ yq_{a^{+}}\left(t\right), & \text{otherwise,} \end{cases} \quad \forall a, a^{+} : a_{h} = a^{+}_{t}, \tag{22}$$

where $\beta_{aa^{+}}$ is supply allocation coefficient, the value of which is related to the proportion of lanes between upstream and downstream sections and the control measures at the intersection, and it is regarded as a fixed constant. The total outflow rate of the section can be measured by the following formula after calculating the outflow rate in each turning direction in the queuing part:

$$yq_{a}\left(t\right) = \sum_{a^{+}: a_{h} = a^{+}_{t}} yq_{aa^{+}}\left(t\right) \quad \forall a. \tag{23}$$

Assume that vehicles in each queue on entrance lane are evenly mixed together, and the outflow rate of the queue on specific route can be calculated by the following formula:

$$yq^{aa^{+}}_{mnr}\left(t\right) = \begin{cases} \dfrac{nq^{aa^{+}}_{mnr}}{nq_{aa^{+}}\left(t\right)} \cdot yq_{aa^{+}}\left(t\right), & \text{if } nq_{aa^{+}}\left(t\right) > 0 \\ \dfrac{xq^{aa^{+}}_{mnr}}{xq_{aa^{+}}} \cdot yq_{aa^{+}}\left(t\right), & \text{otherwise,} \end{cases} \quad \forall m, n, r, \quad \forall r \in R_{mn}, a, a^{+} \in r. \tag{24}$$

The number of vehicles in each queue can be updated by the following formulas:

$$nq\ ^{aa^{+}}_{mnr}\left(t+1\right) = nq^{aa^{+}}_{mnr}\left(t\right) + xq^{aa^{+}}_{mnr}\left(t\right) - yq^{aa^{+}}_{mnr}\left(t\right), \quad \forall m, n, r, \quad \forall r \in R_{mn}, \quad a, a^{+} \in r, \tag{25}$$

$$nq_{aa^{+}}\left(t\right) = \sum_{m \in M}\sum_{n \in N}\sum_{\substack{r: r \in R_{mn} \\ a, a^{+} \in r}} nq^{aa^{+}}_{mnr}\left(t\right) \quad \forall a, a^{+} : a_{h} = a^{+}_{t}, \tag{26}$$

$$nq_{a}\left(t\right) = \sum_{a^{+}: a_{h} = a^{+}_{t}} nq_{aa^{+}}\left(t\right) \quad \forall a. \tag{27}$$

*4.4. Simulation of Evacuation Traffic Flow.* This part is about the simulation process of traffic flow entering and running in the road network with the given evacuation route and departure time. The simulation can transform the state description of the road network based on route into that based on section, and finally contribute to the optimization and evaluation of evacuation organization planning.

According to the optimization model of evacuation route and departure time, simulation inputs are the departure time and running routes of the vehicles (or in each decision period, the number of vehicles that depart along different routes). Assume that vehicles evenly depart from the evacuation source in each decision period. For example, there are 50 vehicles leaving the evacuation source along certain route in 10 minutes, then it can be described as that these 50 vehicles depart evenly in 10 minutes with a flow rate of 5 vehicle/min. The departure rate of the dynamic route can be measured by the following formula:

$$f^{l}_{mnr}(t) = \frac{F^{l}_{mnr}}{\Delta} \quad l_s \le t \cdot \sigma \le l_e, \forall l. \tag{28}$$

In addition, for the given evacuation organization planning $f^{l}_{mnr}(t)$ is included in the following formula:

$$xr^{a}_{mnr}(t) = \begin{cases} f^{l}_{mnr}(t), & \text{if } a \text{ is the first section on route } r \\ yq^{(a-r)a}_{mnr}(t), & \text{otherwise,} \end{cases} \qquad \forall m, n, a, \quad \forall r \in R_{mn}, \tag{29}$$

where $a$-$r$ is the section in front of section $a$ on route $r$.

The steps of the simulation are as follows.

*Step 1.* Let $t = 1$, $xr^{a}_{mnr}(t) = 0$, and $nq^{aa+}_{mnr}(t) = 0$, for all $m, n, t, r$, for all $a$, $a^{+}_{t} = a_h$.

*Step 2.* Calculation and derivation in the running part.

(1) According to the given inputs, $xr^{a}_{mnr}(t)$ can be measured by formulas (28)-(29).

(2) Calculate $nr_a(t)$ by formula (16) and $nq_a(t)$ by formulas (26) and (27).

(3) Calculate $tr_a(t)$ for all $t, a$ by formula (12).

(4) Calculate $yr^{a}_{mnr}(t + tr_a(t))$ and $yr^{a}_{mnr}(j)$, for all $m, n, r, t, a$, $j \in [t + tr_a(t), t + 1 + tr_a(t + 1)]$ and $j$ is an integer by formulas (13)-(14).

(5) Calculate $nr^{a}_{mnr}(t)$ for all $m, n, r, t, a$ by formula (15).

*Step 3.* Calculation and derivation in the queuing part.

(1) Calculate $xq^{aa^{+}}_{mnr}(t)$,for all $m, n, r, t$, for all $a : a^{+}_{t} = a_h$ by formula (17).

(2) Calculate $xq_{aa^{+}}(t), nq_{aa^{+}}(t)$, for all $t, a, a^{+} : a^{+}_{t} = a_h$ by formulas (18) and (26).

(3) Calculate $yq_{aa^{+}}(\mathrm{t})$, $yq_a(t)$, for all $t, a, a^{+} : a^{+}_{t} = a_h$ by formulas (19)–(23).

(4) Calculate $yq^{aa^{+}}_{mnr}(t)$ and $nq^{aa^{+}}_{mnr}(t)$, for all $m, n, r, t$, for all $a : a^{+}_{t} = a_h$ by formulas (24) and (25).

*Step 4.* If $t = M$, then stop; otherwise let $t = t + 1$, and go back to Step 2.

## 5. Model Solution

*5.1. Optimal Conditions Analysis.* Among all the constraints from formulas (2) to (10) shown in the model, what really affect the decision variables are the conservation constraint of total evacuation demand in formula (3) and the nonnegative constraints on the decision variables (or $F^{l}_{mnr} \ge 0$). Actually, the objective of our model is to find the minimum with a series of equation constraints. Thus we can establish the Lagrange function based on formulas (2) and (3):

$$L(F, \theta) = Z(F) + \sum_{m}\sum_{n}\theta_{mn}\left(D_{mn} - \sum_{r}\sum_{l}F^{l}_{mnr}\right), \tag{30}$$

where $\theta$ stands for Lagrange multiplier vector, and $\theta_{mn}$ represents two-dimensional Lagrange multiplier.

According to the Kuhn-Tucker conditions, the Lagrange function above must meet the following conditions at the extreme point:

$$F^{l}_{mnr}\frac{\partial L(F, \theta)}{\partial F^{l}_{mnr}} = 0, \qquad \frac{\partial L(F, \theta)}{\partial F^{l}_{mnr}} \ge 0 \quad \forall m, n, r, l, \tag{31}$$

$$\frac{\partial L(F, \theta)}{\partial \theta_{mn}} = 0 \quad \forall m, n. \tag{32}$$

For certain $F^{s}_{ijw}$, where $i = m$, $j = n$, $w = r$, $s = l$, the partial derivative of the Lagrange function can be calculated as follows:

$$\frac{\partial L(F, \theta)}{\partial F^{s}_{ijw}} = \frac{\partial Z(F)}{\partial F^{s}_{ijw}} + \frac{\partial}{\partial F^{s}_{ijw}}\left[\sum_{m}\sum_{n}\theta_{mn}\left(D_{mn} - \sum_{r}\sum_{l}F^{l}_{mnr}\right)\right] \quad \forall i, j, w, s. \tag{33}$$

There is no relationship between $F^{s}_{ijw}$ and $\theta_{mn}$, also between $F^{s}_{ijw}$ and $D_{mn}$.

What is more,

$$\frac{\partial F^{l}_{mnr}}{\partial F^{s}_{ijw}} = \begin{cases} 1, & \text{if } m = i,\ n = j,\ r = w,\ l = s \\ 0, & \text{others.} \end{cases} \tag{34}$$

Therefore, we can get

$$\frac{\partial}{\partial F^{s}_{ijw}}\left[\sum_{m}\sum_{n}\theta_{mn}\left(D_{mn} - \sum_{r}\sum_{l}F^{l}_{mnr}\right)\right] = -\theta_{ij}. \tag{35}$$

Let

$$\beta^{s}_{ijw}(F) = \frac{\partial Z(F)}{\partial F^{s}_{ijw}}, \tag{36}$$

where $\beta^{s}_{ijw}$ means the marginal evacuation time of the dynamic route between $i$ and $j$ under the traffic condition determined by $F$, that is to say, $\beta^{s}_{ijw}$ is the absolute amount that the value of objective function increases when $F_{ijw}^{s}$ increases one unit.

By substituting formulas (35) and (36) into (33), we can obtain

$$\frac{\partial L(F,\theta)}{\partial F_{ijw}^{s}} = \beta_{ijw}^{s}(F) - \theta_{ij}. \tag{37}$$

Thus formula (31) that describes the Kuhn-Tucker conditions can be written briefly as follows:

$$\begin{gathered} F_{mnr}^{l}\left(\beta_{mnr}^{l}(F) - \theta_{mn}\right) = 0 \quad \forall m,n,r,l, \\ \left(\beta_{mnr}^{l}(F) - \theta_{mn}\right) \geq 0 \quad \forall m,n,r,l. \end{gathered} \tag{38}$$

Formula (38) shows that the marginal evacuation time of used dynamic routes between each evacuation OD is same at the extreme points, where the arrangements for the departure time and evacuation route are the best. As to the unused dynamic routes, their marginal evacuation time is greater than or equal to that of the used. Thus we conclude that the Lagrange multiplier $\theta_{mn}$ is the minimum evacuation time of all the dynamic routes between $m$ and $n$.

### 5.2. Heuristic Algorithm

*5.2.1. Framework and Process of This Algorithm.* In this part, we propose a heuristic algorithm. Once given a set of initial values of the decision variables, the optimal solution can be approximately calculated through continuous iteration and adjustment. Adjustment process is mainly based on the marginal evacuation time, and the goal of the adjustment is to make the marginal evacuation time of used dynamic routes between each evacuation OD equal to the minimum, as well as to make the marginal evacuation time of unused dynamic routes greater than or equal to the minimum marginal evacuation time. The framework and process of the algorithm are shown in Figure 1. The algorithm will not come to an end until it meets certain convergence criteria.

*Step 1.* Give the decision variables a set of initial values. In order to guarantee the feasibility of the initial solution, let

$$\begin{gathered} \left(F_{mnr}^{l}\right)_1 = \frac{D_{mn}}{\left(J \cdot \Delta \cdot \left|R_{mn}\right|\right)}, \quad l = 1,2,\ldots,J, \\ \left(F_{mnr}^{l}\right)_1 = 0, \quad l = J+1, J+2, \ldots, K. \end{gathered} \tag{39}$$

This practice actually distributes the total demand of each evacuation OD to the available dynamic routes evenly, thus ensuring the conservation of the total demand. By choosing an appropriate $J$, we can ensure that all evacuation vehicles will reach the ending node before period $K$. In addition, in order to calculate conveniently, we convert the traffic volume (or the number of vehicles) $F_{mnr}^{l}$ on the dynamic route into departure rate $f_{mnr}^{l}$.

*Step 2.* Calculate marginal evacuation time of the dynamic routes, and the process will be described in detail in Section 5.2.2.

*Step 3-Step 4.* Adjust current solution according to the marginal evacuation time. For each evacuation OD, comparing the marginal evacuation time of each dynamic route and the minimum marginal evacuation time, if the difference between the two is quite great, then reduce the departure rate of the dynamic route appropriately. If the difference between the two is relatively small, then increase the departure rate. In order to ensure the feasibility of the adjusted solution, the reduction should be evenly allocated to the dynamic routes that need to increase the departure rates.

*Step 5.* Judge the convergence: when the difference between the two decision variable values in adjacent iterations is small enough, the algorithm stops.

Step 2–Step 5 are the process of loop iteration.

As we know, the parameter selected plays an important role in the performance of an algorithm. Here we chose appropriate parameter values through numerous trials. And experiments show that the algorithm can surely and rapidly get global optimum solution and greatly increase the accuracy.

In the case analysis shown in Section 6, in two different evacuation demands, we can get the optimum solutions after iterating 60 times and 21 times, respectively, and compared to other algorithms, there is a big promotion in the convergence rate.

*5.2.2. Calculation of Marginal Evacuation Time*

*(1) Derivation of Marginal Evacuation Time.* The calculation of marginal evacuation time is the key step in the algorithm, which is also a bridge between the decision variables and adjusting bases. For the dynamic route $(w,s)$ between certain evacuation OD $(i,j)$, the marginal evacuation time can be calculated by the following formula which is based on formulas (2) and (36):

$$\begin{aligned} \beta_{ijw}^{s}(F) &= \left(l_s + \alpha \cdot \Delta\right) + t_{ijw}^{s}(F) \\ &\quad + \sum_{m\in M}\sum_{n\in N}\sum_{r\in R_{mn}}\sum_{l=1}^{K} F_{mnr}^{l} \cdot \frac{\partial}{\partial F_{ijw}^{s}}\left(t_{mnr}^{l}(F)\right). \end{aligned} \tag{40}$$

The three terms on the right of formula (40), respectively, stand for the three aspects of the absolute amount that the total evacuation time increases when the number of vehicles increases one unit. The first term means the waiting time of the new coming vehicle at evacuation source. The second term represents the running time of the new coming vehicle on the route. The third term stands for the effect exerted by the new coming vehicle on other vehicles' running time. For simplicity, assume that the new coming vehicle has no effect on vehicles in different batches, and we only take the effect on vehicles in same batch into consideration. Thus formula (40) can be written briefly as follows:

$$\beta_{ijw}^{s}(F) = \left(l_s + \alpha \cdot \Delta\right) + t_{ijw}^{s}(F) + F_{ijw}^{s}\frac{\partial}{\partial F_{ijw}^{s}}\left(t_{ijw}^{s}(F)\right), \tag{41}$$

where $l_s + \alpha \cdot \Delta$ is marginal waiting time and the sum of the last two terms is marginal running time. The value of

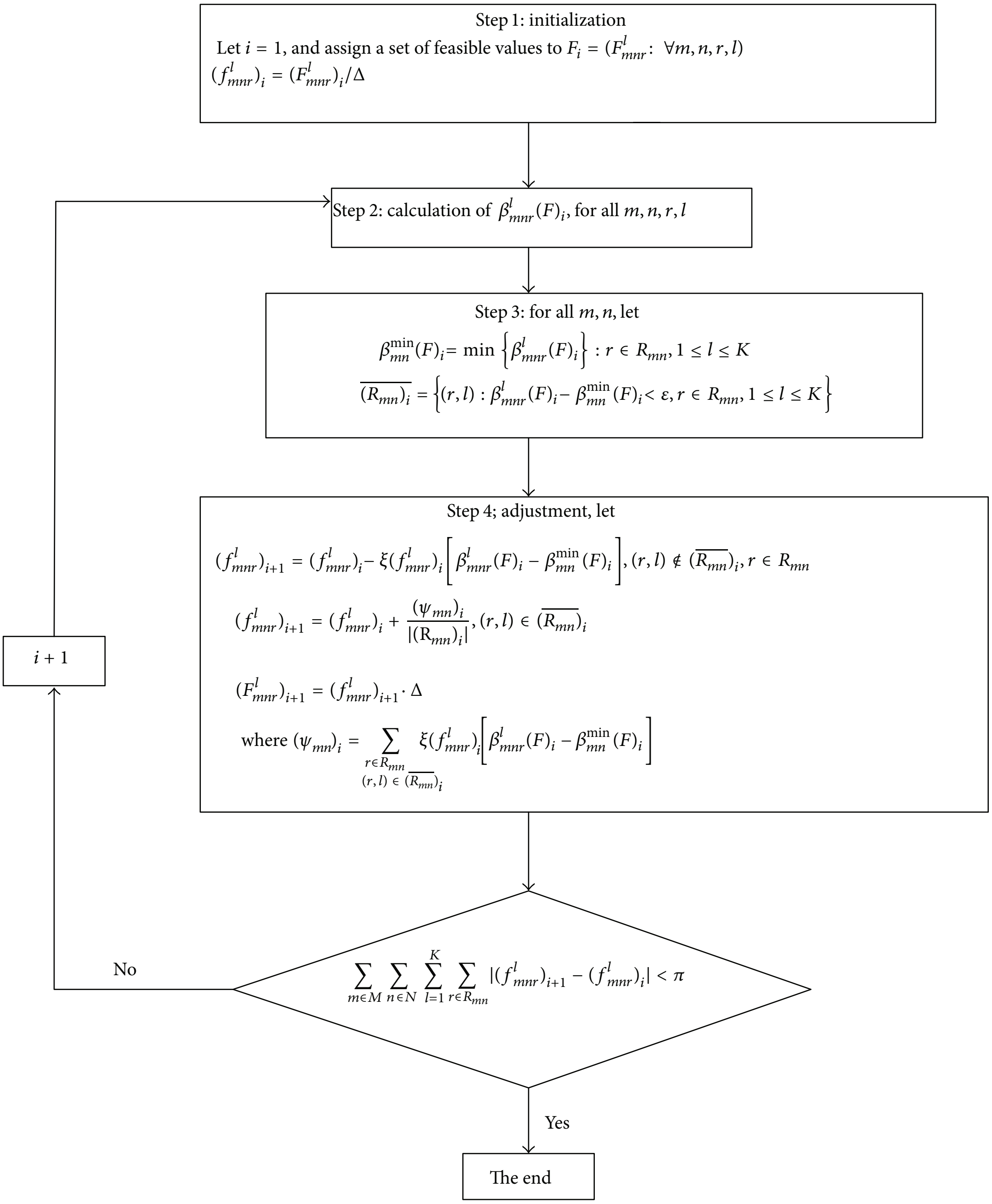

Step 1: initialization
Let $i = 1$, and assign a set of feasible values to $F_i = (F^l_{mnr}: \forall m, n, r, l)$
$(f^l_{mnr})_i = (F^l_{mnr})_i/\Delta$
Step 2: calculation of $\beta^l_{mnr}(F)_i$, for all $m, n, r, l$
Step 3: for all $m, n$, let
$\beta^{\min}_{mn}(F)_i = \min\left\{\beta^l_{mnr}(F)_i\right\} : r \in R_{mn}, 1 \le l \le K$
$\overline{(R_{mn})_i} = \left\{(r,l) : \beta^l_{mnr}(F)_i - \beta^{\min}_{mn}(F)_i < \varepsilon, r \in R_{mn}, 1 \le l \le K\right\}$
Step 4; adjustment, let
$(f^l_{mnr})_{i+1} = (f^l_{mnr})_i - \xi(f^l_{mnr})_i\left[\beta^l_{mnr}(F)_i - \beta^{\min}_{mn}(F)_i\right], (r,l) \notin (\overline{R_{mn}})_i, r \in R_{mn}$
$(f^l_{mnr})_{i+1} = (f^l_{mnr})_i + \frac{(\psi_{mn})_i}{|(\mathrm{R}_{mn})_i|}, (r,l) \in (\overline{R_{mn}})_i$
$(F^l_{mnr})_{i+1} = (f^l_{mnr})_{i+1} \cdot \Delta$
where $(\psi_{mn})_i = \sum_{\substack{r \in R_{mn} \\ (r,l) \in \overline{(R_{mn})}_i}} \xi(f^l_{mnr})_i\left[\beta^l_{mnr}(F)_i - \beta^{\min}_{mn}(F)_i\right]$
$\sum_{m \in M}\sum_{n \in N}\sum_{l=1}^{K}\sum_{r \in R_{mn}} |(f^l_{mnr})_{i+1} - (f^l_{mnr})_i| < \pi$
No
$i + 1$
Yes
The end

Figure 1: Framework of the algorithm.

$t^s_{ijw}(F)$ is related to the traffic condition of the road network in evacuation period, and $F^s_{ijw}$ is determined by the evacuation organization planning, so there exists no analytic function relationship between $t^s_{ijw}(F)$ and $F^s_{ijw}$. It is impossible to calculate the marginal running time of certain dynamic route directly, and we can only calculate it approximately by accumulating the marginal running time of the sections contained in the route.

Assume that route $r$ consists of sections $a_1, a_2, \ldots, a_m$, $\beta^l_{mnr}(F)$ can be calculated by the following formula:

$$\beta^l_{mnr}(F) = t_1 + \beta_{a_1}(t_1) + \beta_{a_2}(t_2) + \cdots + \beta_{a_m}(t_m), \quad (42)$$

where $\beta_{a_i}(t_i)$ means the marginal running time of section $a_i$, namely the absolute amount that the total running time increases when the number of vehicles in section $a_i$ increases

Table 1: Basic parameters setting.

| Section | Running speed<br>$V$ (km/h) | Section length<br>$L$ (m) | Traffic capacity<br>$C$ (c/h) |
|---|---|---|---|
| (1–4) | 10 | 600 | 1900 |
| (1–5) | 10 | 600 | 1100 |
| (2–9) | 15 | 600 | 1200 |
| (2–10) | 13 | 600 | 2100 |
| (6-5) | 15 | 800 | 1900 |
| (5-4) | 14 | 800 | 1850 |
| (4-3) | 16 | 800 | 2000 |
| (10–6) | 9 | 900 | 1700 |
| (5–9) | 13 | 900 | 1100 |
| (4–8) | 16 | 900 | 2200 |
| (3–7) | 10 | 900 | 2300 |
| (10-9) | 10 | 850 | 1850 |
| (9-8) | 15 | 850 | 1950 |
| (8-7) | 13 | 900 | 2100 |
| (10-11) | 12 | 1200 | 1200 |
| (9–12) | 14 | 1000 | 2500 |
| (8–12) | 12 | 1000 | 1800 |
| (7–13) | 13 | 900 | 1050 |
| (11-12) | 15 | 900 | 1970 |
| (12-13) | 10 | 800 | 2050 |

Table 2: Traffic capacity at turning intersections.

| Intersection | Turning direction<br>Node | Traffic capacity<br>(c/h) | Intersection | Turning direction<br>Node | Traffic capacity<br>(c/h) |
|---|---|---|---|---|---|
| 3 | (4-3-7) | 1800 | 9 | (10-9-8) | 1000 |
| 4 | (1-4-3) | 960 | | (10-9-12) | 1100 |
| | (1-4-8) | 1060 | | (2-9-8) | 1200 |
| | (5-4-3) | 940 | | (2-9-12) | 960 |
| | (5-4-8) | 1130 | 10 | (2-10-9) | 540 |
| 5 | (1-5-4) | 600 | | (2-10-11) | 720 |
| | (6-5-4) | 570 | | (6-10-9) | 1120 |
| 6 | (10-6-5) | 1100 | | (6-10-11) | 1200 |
| 7 | (3-7-13) | 1220 | 11 | (10-11-12) | 1400 |
| | (8-7-13) | 1230 | 12 | (8-12-13) | 1200 |
| 8 | (4-8-7) | 560 | | (9-12-13) | 1050 |
| | (4-8-12) | 680 | | (11-12-13) | 1350 |
| | (9-8-7) | 720 | | | |
| | (9-8-12) | 800 | | | |

one unit at time $t_i$. $t_1 = l_s + \alpha \cdot \Delta$ is the departure time of the new coming vehicle that departs from $m$ and $n$ along route $r$ in period $l$, and it is also the time when the vehicle enters section $a_1$, so $t_i$, $i = 2, 3, \ldots, m$, is the time when the vehicle enters section $a_i$. Moreover, $t_i$ depends on the running and propagation of the corresponding evacuation traffic flow in the road network, and $\beta_{a_i}(t_i)$ is determined by the objective function and evacuation traffic flow propagation. So both of the two variables are related to the description method of evacuation traffic flow propagation mechanism.

The running and propagation of the traffic flow in the model are described by a simulation module illustrated in Section 4. The calculation of marginal running time can be carried out in two steps. Firstly, load the corresponding traffic flow into the evacuation network by the simulation module to get the dynamic state of sections in evacuation period that includes the number of vehicles in the sections, inflows and outflows. Then calculate the marginal evacuation time of the dynamic route by accumulating the marginal running time of the sections contained in the route based on formula (42).

TABLE 3: Available routes between each evacuation OD.

| OD | Route number | Route (a sequence of nodes) |
|---|---|---|
| 1-13 | 1 | 1-4-3-7-13 |
| | 2 | 1-4-8-12-13 |
| | 3 | 1-4-8-7-13 |
| | 4 | 1-5-4-3-7-13 |
| | 5 | 1-5-4-8-12-13 |
| | 6 | 1-5-4-8-7-13 |
| | 7 | 1-5-9-8-7-13 |
| | 8 | 1-5-9-8-12-13 |
| | 9 | 1-5-9-12-13 |
| 2-13 | 10 | 2-9-8-7-13 |
| | 11 | 2-9-8-12-13 |
| | 12 | 2-9-12-13 |
| | 13 | 2-10-9-8-7-13 |
| | 14 | 2-10-9-8-12-13 |
| | 15 | 2-10-9-12-13 |
| | 16 | 2-10-11-12-13 |
| | 17 | 2-10-6-5-4-3-7-13 |
| | 18 | 2-10-6-5-4-8-12-13 |
| | 19 | 2-10-6-5-4-8-7-13 |
| | 20 | 2-10-6-5-4-3-7-13 |
| | 21 | 2-10-6-5-5-4-8-12-13 |
| | 22 | 2-10-6-5-5-4-8-7-13 |
| | 23 | 2-10-6-5-5-9-8-7-13 |
| | 24 | 2-10-6-5-5-9-8-12-13 |
| | 25 | 2-10-6-5-5-9-12-13 |

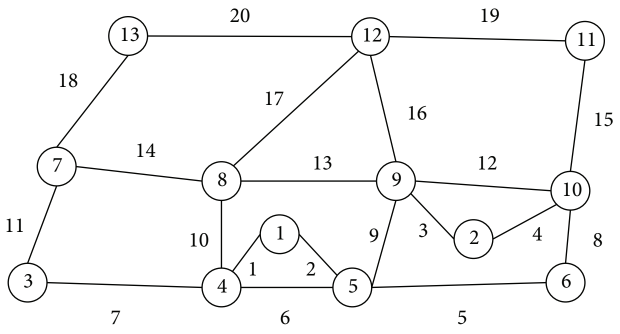


FIGURE 2: Typical evacuation network.

*(2) Calculation of Marginal Running Time.* The dynamic marginal running time of a section can be divided into two parts: the marginal running time in the running part and the marginal delay in the queuing part. The marginal running time in the running part means the absolute amount that the total running time increases when the number of vehicles in the section increase one unit at time $t$, namely, the running time of the new coming vehicle in the section according to the above assumptions, as shown in what follows:

$$\beta_{a_i}(t_i) = t_{a_i}(t_i). \qquad (43)$$

The marginal delay in the queuing part, which includes the delay of the new coming vehicle itself and other vehicles effected by the former, should be respectively calculated according to different turning directions. Here we only take the effects on vehicles in the same queue into consideration, and the marginal delay can be calculated by following formula:

$$\beta_{a_i a_{i+1}}(\tau_i) = \frac{nq_{a_i a_{i+1}}(\tau_i)}{yq_{a_i a_{i+1}}(\tau_i)} + \frac{1}{yq_{a_i a_{i+1}}(\tau_i)} Cnq_{a_i a_{i+1}}(\tau_i), \qquad (44)$$

where $\beta_{a_i a_{i+1}}(\tau_i)$ is the marginal delay of vehicles that enter the turning queue from section $a_i$ to section $a_{i+1}$ at time $\tau_i$. $Cnq_{a_i a_{i+1}}(\tau_i)$ represents the number of vehicles in the turning queue from section $a_i$ to section $a_{i+1}$ influenced by the new coming vehicle at time $\tau_i$. Thus, $Cnq_{a_i a_{i+1}}(\tau_i)$ can be calculated by the following formula:

$$\begin{aligned} & Cnq_{a_i a_{i+1}}(\tau_i) \\ & \quad = xq_{a_i a_{i+1}}\left(\tau_i + \frac{nq_{a_i a_{i+1}}(\tau_i)}{2yq_{a_i a_{i+1}}(\tau_i)}\right) \cdot \frac{nq_{a_i a_{i+1}}(\tau_i)}{yq_{a_i a_{i+1}}(\tau_i)}. \end{aligned} \qquad (45)$$

The marginal evacuation time of route $r$, which consists of sections $a_1, a_2, \ldots, a_m$, can be calculated as follows:

$$\begin{aligned} \beta_{mnr}^{l}(F) = {} & t_1 + \beta_{a_1}(t_1) + \beta_{a_1 a_2}(\tau_2) + \beta_{a_2}(t_2) \\ & + \beta_{a_2 a_3}(\tau_2) + \cdots + \beta_{a_{m-1} a_m}(\tau_{m-1}) + \beta_{a_m}(t_m), \end{aligned} \qquad (46)$$

where $t_i$ and $\tau_i$ $(i = 2, 3, \ldots, m-1)$, respectively, represent the time when the new coming vehicle that departs along route $r$ in period $l$ enters section $a_i$, and the time when this vehicle enters the queuing part after leaving section $a_i$. The two time variables can be measured by formula (47):

$$\begin{gathered} \tau_i = t_i + t_{a_i}(t_i), \\ t_{i+1} = \tau_i + \frac{nq_{a_i a_{i+1}}(\tau_i)}{yq_{a_i a_{i+1}}(\tau_i)} \\ i = 2, 3, \ldots, m-1, \end{gathered} \qquad (47)$$

where $nq_{a_i a_{i+1}}(\tau_i)/yq_{a_i a_{i+1}}(\tau_i)$ is used to calculate the delay of the vehicles that enter the turning queue from section $a_i$ to section $a_{i+1}$ after finishing section $a_i$ at time $\tau_i$.

## 6. Case Analysis

In this section, we will analyze the evacuation in areas of Beijing. The evacuation routes in Beijing have the following typical characteristics: on the whole, all of the evacuation routes look like a checkerboard, and generally there are crisscrossed small road networks around the evacuation sources. In this way, we can draw a typical evacuation road network as shown in Figure 2.

Table 4: Optimal evacuation planning ($D_{1,13} = 1000$, $D_{2,13} = 1500$).

| | | Departure time (every 6 minutes) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| OD | Route | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1-13 | 1 | 59 | 66 | 83 | 96 | | | | |
| | 2 | 40 | 58 | 51 | 50 | | | | |
| | 3 | 22 | 32 | 37 | 37 | | | | |
| | 4 | 22 | 34 | 32 | 8 | | | | |
| | 5 | 7 | 7 | 6 | 2 | | | | |
| | 6 | 8 | 12 | 12 | 3 | | | | |
| | 7 | 12 | 18 | 16 | 4 | | | | |
| | 8 | 8 | 35 | 39 | 11 | | | | |
| | 9 | 20 | 20 | 23 | 18 | | | | |
| 2-13 | 10 | 40 | 53 | 75 | 50 | 15 | 19 | 12 | 6 |
| | 11 | — | 10 | 7 | 17 | 12 | 6 | | |
| | 12 | 16 | 17 | 15 | 21 | 13 | 8 | 2 | |
| | 13 | 8 | 11 | 12 | 14 | 5 | 6 | | |
| | 14 | 31 | 18 | 37 | 44 | 8 | | | |
| | 15 | 12 | 15 | 16 | 17 | 3 | 5 | | |
| | 16 | 8 | 14 | 12 | 7 | | | | |
| | 17 | 30 | 11 | 30 | 25 | 22 | 6 | | |
| | 18 | 5 | 55 | 68 | 73 | | | | |
| | 19 | 14 | 8 | 12 | | | | | |
| | 20 | 16 | 8 | | | | | | |
| | 23 | 4 | | | | | | | |

Table 5: Optimal evacuation planning ($D_{1,13} = 300$, $D_{2,13} = 500$).

| | Departure time (every 6 minutes) | | | |
|---|---|---|---|---|
| OD | Route | 1 | 2 | 3 |
| 1-13 | 1 | 80 | 29 | 9 |
| | 2 | 52 | 16 | 3 |
| | 3 | 20 | 13 | 6 |
| | 4 | 10 | 7 | 3 |
| | 5 | 8 | 6 | 5 |
| | 6 | 16 | 5 | 7 |
| | 8 | 13 | 4 | 5 |
| | 9 | 10 | 7 | 2 |
| 2-13 | 10 | 80 | 50 | 5 |
| | 11 | 6 | 3 | 1 |
| | 12 | 6 | 6 | — |
| | 13 | 30 | 20 | 6 |
| | 14 | 35 | 7 | 2 |
| | 15 | 72 | 45 | 13 |
| | 16 | 15 | 4 | 5 |

The basic parameters setting, traffic capacity at turning intersections, and available routes between each evacuation OD are, respectively, shown in Tables 1, 2, and 3. For the sake of simplicity, let $t = 6$, $\triangle = 6$ (min), $\xi = 0.01$, $\varepsilon = 1.98$, and convergence parameter $\pi = 1.532$.

By the aforementioned model and algorithm, we can get the optimal evacuation organization planning (shown in Tables 4 and 5) corresponding to two different evacuation demands after iterating 60 times and 21 times, and the parameters of the two demands are ($D_{1,13} = 1000$, $D_{2,13} = 1500$), ($D_{1,13} = 300$, $D_{2,13} = 500$). Moreover, the number of vehicles that depart along different routes are shown in the tables and the respective evacuation time is 59 minutes and 25 minutes.

## 7. Conclusions

Due to the basic features of major incident and characteristics in urban area, once a major incident occurs, it subsequently leads to great losses. Therefore, the government needs to improve its ability for emergency management. This paper establishes a comprehensive model for the optimization of emergency evacuation route and departure time based on the simulation. In the process, we find reasonable description method of evacuation traffic flow propagation under traffic control, and also propose a heuristic algorithm for the optimization of this comprehensive model. In case analysis, we take some areas in Beijing as evaluation sources to verify the reliability of this model. All in all, our evaluation model can be applied to the actual situation under traffic control.

## 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 National Natural Science Foundation of China (nos. 71102174 and 71372019), Beijing Higher Education Young Elite Teacher Project (no. YETP1173), Specialized Research Fund for Doctoral Program of Higher Education of China (no. 20111101120019), and Beijing Philosophy and Social Science Foundation of China (no. 11JGC106).

## References

[1] A. Desmet and E. Gelenbe, "Graph and analytical models for emergency evacuation," *Future Internet*, vol. 5, no. 1, pp. 46–55, 2013.

[2] A. M. Caunhye, X. Nie, and S. Pokharel, "Optimization models in emergency logistics: a literature review," *Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 4–13, 2012.

[3] A. J. Pel, M. C. J. Bliemer, and S. P. Hoogendoorn, "A review on travel behaviour modelling in dynamic traffic simulation models for evacuations," *Transportation*, vol. 39, no. 1, pp. 97–123, 2012.

[4] Y. M. Chen and D. Y. Xiao, "Emergency evacuation model and algorithms," *Journal of Transportation Systems Engineering and Information Technology*, vol. 8, no. 6, pp. 96–100, 2008.

[5] E. Kwon and S. Pitt, "Evaluation of emergency evacuation strategies for downtown event traffic using a dynamic network model," *Transportation Research Record*, no. 1922, pp. 149–155, 2005.

[6] T. J. Cova and J. P. Johnson, "A network flow model for lane-based evacuation routing," *Transportation Research A*, vol. 37, no. 7, pp. 579–604, 2003.

[7] U. Petruccelli, "Urban evacuation in seismic emergency conditions," *ITE Journal*, vol. 73, no. 8, pp. 34–38, 2003.

[8] D. J. Zawack and G. L. Thompson, "A dynamic space-time network flow model for city traffic congestion," *Transportation Science*, vol. 21, no. 3, pp. 153–162, 1987.

[9] B. Yang, Y. G. Wu, and B. Ren, "Application of multi-resolution modelling in emergency evacuation simulation," *Simulation and Process Modeling*, vol. 7, no. 1-2, pp. 89–97, 2012.

[10] N. Pelechano and A. Malkawi, "Evacuation simulation models: challenges in modeling high rise building evacuation with cellular automata approaches," *Automation in Construction*, vol. 17, no. 4, pp. 377–385, 2008.

[11] M. Jha, K. Moore, and B. Pashaie, "Emergency evacuation planning with microscopic traffic simulation," *Transportation Research Record*, no. 1886, pp. 40–48, 2004.

[12] Y. C. Chiu, H. Zheng, J. Villalobos, and B. Gautam, "Modeling no-notice mass evacuation using a dynamic traffic flow optimization model," *IIE Transactions*, vol. 39, no. 1, pp. 83–94, 2007.

[13] M. Pidd, F. N. De Silva, and R. W. Eglese, "A simulation model for emergency evacuation," *European Journal of Operational Research*, vol. 90, no. 3, pp. 413–419, 1996.

[14] M. Hasby and M. L. Khodra, "Optimal path finding based on traffic information extraction from Twitter," in *Proceedings of the IEEE International Conference ICT for Smart Society (ICISS '13)*, pp. 1–5, June 2013.

[15] L. Wu and H. Lin, "A case study of developing personalized spatial cognitive road network and raster capable route finding algorithm for pedestrian evacuation behavior simulation," in *Proceedings of the 17th IEEE International Conference on Geoinformatics*, pp. 1–6, August 2009.

[16] D. Li, X. Zhang, and L. Wang, "On the crowded places multi-exits emergency evacuation model and algorithm," *International Journal of Computer Science Issues*, vol. 10, no. 3, pp. 188–191, 2013.

[17] Q. Lu, B. George, and S. Shekhar, "Capacity constrained routing algorithms for evacuation planning: a summary of results," *Advances in Spatial and Temporal Databases*, vol. 3633, pp. 291–307, 2005.

[18] H. D. Sherali, T. B. Carter, and A. G. Hobeika, "A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions," *Transportation Research Part B*, vol. 25, no. 6, pp. 439–452, 1991.

[19] V. Papinigisa, E. Gedabcand, and K. Lukošius, "Design of people evacuation from rooms and buildings," *Journal of Civil Engineering and Management*, vol. 16, no. 1, pp. 131–139, 2010.

[20] C. E. Dunn and D. Newton, "Optimal routes in GIS and emergency planning applications," *Area*, vol. 24, no. 3, pp. 259–267, 1992.

[21] T. Yamada, "A network flow approach to a city emergency evacuation planning," *International Journal of Systems Science*, vol. 27, no. 10, pp. 931–936, 1996.

[22] V. B. G. Campos, P. A. L. da Silva, and P. O. B. Netto, "Evacuation transportation planning: a method of identify optimal independent routes," in *Proceedings of the 5th International Conference on Urban Transport*, pp. 555–564, 2000.

[23] S. Mamada, K. Makino, and S. Fujishige, "The evacuation problem, dynamic network flows, and algorithms," in *Proceedings of the SICE Annual Conference 2003*, pp. 2807–2811, Fukui University, Fukui, Japan, August 2003.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 497941, 21 pages
http://dx.doi.org/10.1155/2014/497941

*Research Article*

# Modeling Markov Switching ARMA-GARCH Neural Networks Models and an Application to Forecasting Stock Returns

**Melike Bildirici[1] and Özgür Ersin[2]**

[1] *Yıldız Technical University, Department of Economics, Barbaros Bulvari, Besiktas, 34349 Istanbul, Turkey*
[2] *Beykent University, Department of Economics, Ayazağa, Şişli, 34396 Istanbul, Turkey*

Correspondence should be addressed to Melike Bildirici; melikebildirici@gmail.com

Received 20 August 2013; Accepted 4 November 2013; Published 6 April 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

The study has two aims. The first aim is to propose a family of nonlinear GARCH models that incorporate fractional integration and asymmetric power properties to MS-GARCH processes. The second purpose of the study is to augment the MS-GARCH type models with artificial neural networks to benefit from the universal approximation properties to achieve improved forecasting accuracy. Therefore, the proposed Markov-switching MS-ARMA-FIGARCH, APGARCH, and FIAPGARCH processes are further augmented with MLP, Recurrent NN, and Hybrid NN type neural networks. The MS-ARMA-GARCH family and MS-ARMA-GARCH-NN family are utilized for modeling the daily stock returns in an emerging market, the Istanbul Stock Index (ISE100). Forecast accuracy is evaluated in terms of MAE, MSE, and RMSE error criteria and Diebold-Mariano equal forecast accuracy tests. The results suggest that the fractionally integrated and asymmetric power counterparts of Gray's MS-GARCH model provided promising results, while the best results are obtained for their neural network based counterparts. Further, among the models analyzed, the models based on the Hybrid-MLP and Recurrent-NN, the MS-ARMA-FIAPGARCH-HybridMLP, and MS-ARMA-FIAPGARCH-RNN provided the best forecast performances over the baseline single regime GARCH models and further, over the Gray's MS-GARCH model. Therefore, the models are promising for various economic applications.

## 1. Introduction

In the light of the significant improvements in the econometric techniques and in the computer technologies, modeling the financial time series have been subject to accelerated empirical investigation in the literature. Accordingly, following the developments in the nonlinear techniques, analyses focusing on the volatility in financial returns and economic variables are observed to provide significant contributions. It could be stated that important steps have been taken in terms of nonlinear measurement techniques focusing on the instability or stability occurring vis-a-vis encountered volatility. Further, the determination of stability or instability in terms of volatility in the financial markets gains importance especially for analyzing the risk encountered. In addition to impact of the magnitude and the size of shocks on volatility, the financial returns are under the influence of sudden or abrupt changes in the economy. Hence, the volatility of economic data has been explored in econometric literature as a result of the need of modelling uncertainty and risk in the financial returns. The relationship between the financial returns and various important factors such as the trade volume, market price of financial assets, and the relationship between volatility, trade volume, and financial returns have been vigorously investigated [1–4].

The ARCH model introduced by Engle [5] and the Generalized ARCH (GARCH) model introduced by Bollerslev [6] are generally accepted for measuring volatility in financial models. GARCH models have been used intensively in academic studies. A tremendous amount of GARCH models exist and various studies provide extended evaluation of the development.

Among many, Engle and Bollerslev [7] developed the Integrated-GARCH (I-GARCH) process to incorporate integration properties, AGARCH model, introduced by Engle [8], allows modeling asymmetric effects of negative and positive innovations. In terms of modeling asymmetries, GARCH models have been further developed by including asymmetric impacts of the positive and negative shocks to capture the asymmetric effects of shocks on volatility and return series which depend on the type of shocks, i.e. either negative or positive. Following the generalization of EGARCH model of Nelson [9] that allows modelling the asymmetries in the relationship between return and volatility, the Glosten et al. [10] noted the importance of asymmetry caused by good and bad news in volatile series and proposed a model that incorporates the past negative and positive innovations with an identity function that leads the conditional variance to follow different processes due to asymmetry. The finding is a result of the empirical analyses which pointed at the fact that the negative shocks had a larger impact on volatility. Consequently, the bad news have a larger impact compared to the conditional volatility dynamics followed after the good news. Due to this effect, asymmetric GARCH models have rapidly expanded. The GJR-GARCH model was developed independently by Zakoian [11, 12] and Glosten et al. [10]. It should be noted that, in terms of asymmetry, the Threshold GARCH (T-GARCH) of Zakoian [12], VGARCH, and nonlinear asymmetric GARCH models (NAGARCH) of Engle and Ng [13] are closely related versions to model asymmetry in financial asset returns. The SQR-GARCH model of Heston and Nandi [14] and the Aug-GARCH model developed by Duan [15] nest several versions of the models taking asymmetry discussed above. Further, models such as the Generalized Quadratic GARCH (GQARCH) model of Sentana [16] utilize multiplicative error terms to capture volatility more effectively. The FIGARCH model of Baillie et al. [17] benefits from an ARFIMA type fractional integration representation to better capture the long-run dynamics in the conditional variance (See for detailed information, Bollerslev [18]). The APARCH/APGARCH model of Ding et al. [19] is an asymmetric model that incorporates asymmetric power terms which are allowed to be estimated directly from the data. The APGARCH model also nests several models such as the TGARCH, TSGARCH, GJR, and logGARCH. The FIAPGARCH model of Tse [20] combines the FIGARCH and the APGARCH. Hyperbolic GARCH (HYGARCH) model of Davidson [21] nests the ARCH, GARCH, IGARCH, and FIGARCH models (for an extended review GARCH models, see Bollerslev [18]).

Even though the ARCH/GARCH models can be applied quickly for many time series, the shortcomings in these models were discussed by certain studies. Perez-Quiros and Timmermann [22] focused on the conditional distributions of financial returns and showed that recessionary and expansionary periods possess different characteristics, while the parameters of a GARCH model are assumed to be stable for the whole period. Certain studies discussed the high volatility persistence inherited in the baseline GARCH and proposed early signs of regime switches. Diebold [23] and Lamoureux and Lastrapes [24] are two of the highly cited studies discussing high persistence in volatility due to structural changes. Lamoureux and Lastrapes [24] showed that the encountered high persistence in volatility processes resulted from the volume effects that had not been taken into account. Qiao and Wong [25] followed a bivariate approach and confirmed that the Lamoureux and Lastrapes [24] effect exists due to the volume and turnover effects on conditional volatility and after the introduction of volume/turnover as exogenous variables, it is possible to obtain a significant decline in the the persistence. Mikosch and Stărică [26] showed that structural changes had an important impact that leads to accepting an integrated GARCH process. Bauwens et al. [27, 28] discussed that the persistence in the estimated single regime GARCH processes could be considered as resulting from the misspecification which could be controlled by introducing an MS-GARCH specification where the regime switches are governed by a hidden Markov chain.

Krämer [29] evaluated the autocorrelation in the squared error terms and provided an important contribution. Accordingly, the observed empirical autocorrelations of the $\varepsilon_t^2$ are much larger than the theoretical autocorrelations implied by the estimated parameters through evaluating an MS-GARCH model where the autocorrelation problem could be shown to accelerate as the transition probabilities approached 1. (For a proof see e.g., Francq and Zakoïan [30]; Krämer [29]) (In particular, the empirical autocorrelations of the $\varepsilon_t^2$ often seem to indicate long memory, which is not possible in the GARCH-model; in fact, in all standard GARCH-models, theoretical autocorrelations must eventually decrease exponentially, so long memory is ruled out). Alexander and Lazaar [31] showed that leverage effects are due to asymmetry in the volatility responses to the price shocks and the leverage effect accelerates once the markets are in the more volatile regime Krämer and Tameze [32] showed that a single state GARCH model had only one mean reversion while by allowing regime switching in the GARCH processes, mean reverting effect diminishes. In a perspective of volatility, if these shifts are persistent, then there are two sources of volatility persistence, due to shocks and due to regime-switching in the parameters of the variance process. By utilizing a Markov transformation model, it could be shown that the relationships among the regimes between the periods of $t-1$ and $t$ could be explained and the most important advantage of the MSGARCH model exposes itself as there is no need for the researchers to observe the regime changes. The model allows different regimes to reveal by itself [33].

The regime switching in light of the Markov switching model has interesting properties to be examined such as the stationarity by allowing the switching course of volatility inherent in the asset prices. The hidden Markov model (HMM) developed by Taylor [34] is a switching model that benefits from including an unobserved variable to capture volatility to be modeled with transitions between the hidden states that possess different probability distributions attached to each state. Hidden Markov model has been applied successfully by Alexander and Dimitriu [35], Cheung and Erlandsson [36], Francis and Owyang [37], and by Clarida et al. [38] to capture the switching type of predictions in stock returns, interest rates, and exchange rates. Regime switching model has been used extensively for prediction of returns belonging to different stock market returns in different economies and by following the fact that the stock market indices are very sensitive to stock volatility, which accelerates especially during periods with market turbulences (see for detailed information, Alexander and Kaeck, [39]).

The conventional statistical techniques for forecasting reached their limit in applications with nonlinearities, furthermore, recent results suggest that nonlinear models tend to perform better in models for stock returns forecasting [40]. For this reason, many researchers have used artificial neural networks methodologies for financial analysis on the stock market. Lai and Wong [41] contributed to the nonlinear time series modeling methodology by making use of single-layer neural network. Further, modeling of NN models for estimation and prediction for time series have important contributions. Weigend et al. [42], Weigend and Gershenfeld [43], White [44], Hutchinson et al. [45], and Refenes et al. [46] contributed to financial analyses, stock market returns estimation, pattern recognition, and optimization. NN modeling methodology is applied successfully by Wang et al. [47] and Wang [48] to forecast the value of stock indices. Similarly, Abhyankar et al. [49], Castiglione [50], Freisleben [51], Kim and Chun [52], Liu and Yao [53], Phua et al. [54], Refenes et al. [55], Resta [56], R. Sitte and J. Sitte [57], Tiňo et al. [58], Yao and Poh [59], and Yao and Tan [60] are important investigations focusing on the relationships between stock prices and market volumes and volatility. For similar applications, see [1–4]. Bildirici and Ersin [61] modeled NN-GARCH family models to forecast daily stock returns for short and long run horizons and they showed that GARCH models augmented with artificial neural networks (ANN) architectures and algorithms provided significant forecasting performances. Ou and Wang [62] extended the NN-GARCH models to Support Vector Machines. Azadeh et al. [63] evaluated NN-GARCH models and proposed the integrated ANN models. Bahrammirzaee [64] provided an analysis based on financial markets to evaluate the artificial neural networks, expert systems, and hybrid intelligence systems. Further, Kanas and Yannopoulos [65] and Kanas [40] used Markov switching and Neural Networks techniques for forecasting stock returns; however, their approaches depart from the approach followed within this study.

In this study, the neural networks and Markov switching structures are aimed to be integrated to augment the ARMA-GARCH models by incorporating regime switching and different neural networks structures. The approach aims at formulations and estimations of MS-ARMA-GARCH-MLP, MS-ARMA-APGARCH-MLP, MS-ARMA-FIGARCH-MLP, MS-ARMA-FIAPGARCH-MLP, MS-ARMA-GARCH-RBF MS-ARMA-APGARCH-RBF, MS-ARMA-FIGARCH-RBF, and MS-ARMA-FIAPGARCH-RBF; the recurrent neural network augmentations of the models are, namely, the MS-ARMA-GARCH-RNN MS-ARMA-APGARCH-RNN, MS-ARMA-FIGARCH-RNN, and MS-ARMA-FIAPGARCH-RNN. And lastly, the paper aims at providing Hybrid NN versions: the MS-ARMA-GARCH-HybridNN, MS-ARMA-APGARCH-HybridNN, MS-ARMA-FIGARCH-HybridNN, and MS-ARMA-FIAPGARCH-HybridNN.

## 2. The MS-GARCH Models

Over long periods, there are many reasons why financial series exhibit important breaks in behavior; examples include depression, recession, bankruptcies, natural disasters, and market panics, as well as changes in government policies, investor expectations, or the political instability resulting from regime change.

Diebold [23] provided a throughout analysis on volatility models. One of the important findings is the fact that volatility models that fail to adequately incorporate nonlinearity are subject to an upward bias in the parameter estimates which results in strong forms of persistence that occurs especially in high volatility periods in financial time series. As a result of the bias in the parameter estimates, one important result of this fact is on the out-of-sample forecasts of single regime type GARCH models. Accordingly, Schwert [66] proposed a model that incorporates regime switching that is governed by a two state Markov process, hence the model retains different characteristics in the regimes that are defined as high volatility and low volatility regimes.

Hamilton [67] proposed the early applications of HMC models within a Markov switching framework. Accordingly, MS models were estimated by maximum likelihood (ML) where the regime probabilities are obtained by the proposed Hamilton-filter [68–71]. ML estimation of the model is based on a version of the Expectation Maximization (EM) algorithm as discussed in Hamilton [72], Krolzig [73–76]. In the MS models, regime changes are unobserved and are a discrete state of a Markov chain which governs the endogenous switches between different AR processes throughout time. By inferring the probabilities of the unobserved regimes which are conditional on an information set, it is possible to reconstruct the regime switches [77].

Furthermore, certain studies aimed at the development of modeling techniques which incorporate both the probabilistic properties and the estimation of a Markov switching ARCH and GARCH models. A condition for the stationarity of a *natural* path-dependent Markov switching GARCH model as in Francq et al. [78] and a throughout analysis of the probabilistic structure of that model, with conditions for the existence of moments of any order, are developed and investigated in Francq and Zakoïan [30]. Wong and Li [79], Alexander and Lazaar [80], and Haas et al. [81–83] derived stationarity analysis for some mixing models of conditional heteroskedasticity [27, 28]. For the Markov switching GARCH models that avoid the dependency of the conditional variance on the chain's history, the stationarity conditions are known for some special cases in the literature [84]. Klaassen [85] developed the conditions for stationarity of the model as the special cases of the two regimes. A necessary and sufficient stationarity condition has been developed by Haas et al. [81–83] for their Markov switching GARCH model. Furthermore, Cai [86] showed the properties of Bayesian estimation of a Markov switching ARCH model where only the constant in the ARCH equation is allowed to have regime switches. The approach has been investigated by Kaufman and Frühwirth-Schnatter [87] and Kaufmann and Scheicher [88]. Das and Yoo [89] proposed an MCMC algorithm for the same model (switches being allowed in the constant term) with a single state GARCH term to show that gains could be achieved to overcome path-dependence. MS-GARCH models are studied by Francq and Zakoïan [30] to achieve their non-Bayesian estimation properties in light of the generalized method of moments. Bauwens et al. [27, 28] proposed a Bayesian Markov chain Monte Carlo (MCMC) algorithm that is differentiated by including the state variables in the parameter space to control the path-dependence by obtaining the parameter space with Gibbs sampling [90].

The high and low volatility probabilities of MS-GARCH models allow differentiating high and low volatility periods. By observing the periods in which volatility is high, it is possible to investigate the economic and political reasons that caused increased volatility. If a brief overview is to be presented, there are several models based on the idea of regime changes which should be mentioned. Schwert [66] explores a model in which switches between these states that returns can have a high or low variance are determined by a two-state Markov process. Lamoureux and Lastrapes [24] suggest the use of Markov switching models for a way of identifying the timing of the shifts in the unconditional variance. Hamilton and Susmel [91] and Cai [86] proposed Markov switching ARCH model to capture the effects of sudden shifts in the conditional variance. Further, Hamilton and Susmel [91] extended the analysis to a model that allows three regimes, which were differentiated between low, moderate and high volatility regimes, where the high-volatility regime captured the economic recessions. It is accepted that the proposals of Cai [86] and Hamilton and Susmel [91] helped the researchers to control for the problem of path dependence, which makes the computation of the likelihood function impossible (The conditional variance at time *t* depends on the entire sequence of regimes up to time *t* due to the recursive nature of the GARCH process. In Markov switching model, the regimes are unobservable, one needs to integrate over all possible regime paths. The number of possible paths grows exponentially with *t*, which renders ML estimation intractable.) (see for detail, Bauwens, et al. [27, 28]).

Gray [92] study is one of the important studies where a Markov switching GARCH model is proposed to overcome the path dependence problem. According to Gray's model, once the conditional volatility processes are differentiated between regimes, an aggregation of the conditional variances for the regimes could be used to construct a single variance coefficient to evaluate the path dependence. A modification is also conducted by Klaassen [85]. Yang [93], Yao and Attali [94], Yao [95], and Francq and Zakoïan [96] derived conditions for the asymptotic stationarity of some AR and ARMA models with Markov switching regimes. Haas et al. [81–83] investigated a MS-GARCH model by which a finite state-space Markov chain is assumed to govern the ARCH parameters, whereas the autoregressive process followed by the conditional variance is subject to the assumption that past conditional variances are in the same regime (for details, the readers are referred to Bauwens et al. [27, 28], Klaassen [85], Haas et al. [81–83], Francq and Zakoïan [30], Krämer [29], and Alexander and Kaeck [39]).

Another area of analysis pioneered by Haas [97] and Chang et al. [98] allow different distributions in order to gain forecast accuracy. An important finding of these studies showed that by allowing the regime densities to follow skew-normal distribution with Gaussian tail characteristics, several return series could be modeled more efficiently in terms of forecast accuracy. Liu [99] developed and discussed the conditions for stationarity in Markov switching GARCH structure in Haas et al. [81–83] and proved the existence of the moments. In addition, Abramson and Cohen [100] discussed and further evaluated the stationarity conditions in a Markov switching GARCH process and extended the analysis to a general case with $m$-state Markov chains and GARCH$(p, q)$ processes. An evaluation and extension of the stationarity conditions for a class of nonlinear GARCH models are investigated in Abramson and Cohen [100]. Francq and Zakoïan [30] derived the conditions for weak stationarity and existence of moments of any order MS-GARCH model. Bauwens et al. [27, 28] showed that by enlarging the parameter space to include space variables, though maximum likelihood estimation is not feasible, the Bayesian estimation of the extended process is feasible for a model where the regime changes are governed with a hidden Markov chain. Further, Bauwens et al. [27, 28] accepted mild regularity conditions under which the Markov chain is geometrically ergodic and has finite moments and is strictly stationary.

### 2.1. MS-ARMA-GARCH Models.

To avoid path-dependence problem, Gray [92] suggests integrating out the unobserved regime path in the GARCH term by using the conditional expectation of the past variance. Gray's MS-GARCH model is represented as follows:

$$\begin{aligned}\sigma^2_{t,(s_t)} &= w_{(s_t)} + \sum_{i=1}^{q} \alpha_{i,(s_t)} \varepsilon^2_{t-i} + \sum_{j=1}^{p} \beta_{j,(s_t)} E\left(\frac{\varepsilon^2_{t-j}}{I_{t-j-1}}\right) \\ &= w_{(s_t)} + \sum_{i=1}^{q} \alpha_{i,(s_t)} \varepsilon^2_{t-i} \\ &\quad + \sum_{j=1}^{p} \beta_{j,(s_t)} \sum_{s_{t-j=1}}^{m} P\left(s_{t-j} = \frac{s_{t-j}}{I_{t-j-1}}\right) \sigma^2_{t-j}, s_{t-j},\end{aligned} \tag{1}$$

where $w_{st} > 0$, $\alpha_{i,st} \geq 0$, $\beta_{j,st} \geq 0$, and $i = 1, \ldots, q$, $j = 1, \ldots, p$, $s_t = 1, \ldots, m$. The probabilistic structure of the switching regime indicator $s_t$ is defined as a first-order Markov process with constant transition probabilities $\pi_1$ and $\pi_2$, respectively ($\Pr\{s_t = 1 \mid s_{t-1} = 1\} = \pi_1$, $\Pr\{s_t = 2 \mid s_{t-1} = 1\} = 1 - \pi_1$, $\Pr\{s_t = 2 \mid s_{t-1} = 2\} = \pi_2$, and $\Pr\{s_t = 1 \mid s_{t-1} = 2\} = 1 - \pi_2$).

Although Dueker [101] accepts a collapsing procedure of Kim's [102] algorithm to overcome path-dependence problem, Dueker [101] adopts the same framework of Gray [92]. Accordingly, the modified GARCH version of Dueker [101] is accepted which governs the dispersion instead of traditional GARCH(1,1) specification.

Yang [103], Yao and Attali [94], Yao [95], and Francq and Zakoïan [96] derived conditions for the asymptotic stationarity of models with Markov switching regimes (see for detailed information Bauwens and Rombouts [104]. The major differences between Markov switching GARCH models are the specification of the *variance process*; that is, the conditional variance $\sigma_t^2 = \mathrm{Var}(\varepsilon_t/S_t)$. To consider the conditional variance as in the Bollerslev's [105] GARCH model and to consider the regime dependent equation for the conditional variance in Frömmel [106] are accepted that The coefficients $w_{st}, \alpha_{st}, \beta_{st}$ correspond to respective coefficients in the one-regime GARCH model, but may differ depending on the present state.

Klaassen [85] (Klassen [85] model is defined as $\sigma^2_{t,(s_t)} = w_{(s_t)} + \sum_{i=1}^{q} \alpha_{i,(s_t)} \varepsilon^2_{t-i} + \sum_{j=1}^{p} \beta_{j,(s_t)} \sum_{\widetilde{s}=1}^{m} P(S_{t-j} = s_{t-j} \mid I_{t-1}, S_t = s_t) \sigma^2_{t-j}, s_{t-j})$ suggested to use the conditional expectation of the lagged conditional variance with a broader information set than the model derived in Gray [92]. Accordingly, Klaassen [85] suggested modifying Gray's [92] model by replacing $p(s_{t-j} = s_{t-j} \mid I_{t-j-1})$ by $p(s_{t-j} = s_{t-j} \mid I_{t-1}, S_t = s_t)$ while evaluating $\sigma_t^2, s_t$.

Another version of MS-GARCH model is developed by Haas et al. [81–83]. According to this model, Markov chain controls the ARCH parameters at each regime $(w_s, \alpha_{i,s})$ and the autoregressive behavior in each regime is subject to the assumption that the past conditional variances are in the same regime as that of the current conditional variance [100].

In this study, models will be derived following the MS-ARMA-GARCH specification in the spirit of Blazsek and Downarowicz [107] where the properties of MS-ARMA-GARCH processes were derived following Gray [92] and Klaassen [85] framework. Henneke et al. [108] developed an approach to investigate the model derived in Francq et al. [78] for which the Bayesian framework was derived. The stationarity of the model was evaluated by Francq and Zakoïan [96] and an algorithm to compute the Bayesian estimator of the regimes and parameters was developed. It should be noted that the MS-ARMA-GARCH models in this paper were developed by following the models developed in the spirit of Gray [92] and Klaassen [85] similar to the framework of Blazsek and Downarowicz [107].

The MS-ARMA-GARCH model with regime switching in the conditional mean and variance are defined as a regime switching model where the regime switches are governed by an unobserved Markov chain in the conditional mean and in the conditional variance processes as

$$\begin{aligned} y_t &= c_{(s_t)} + \sum_{i=1}^{r} \theta_{i,(s_t)} y_{t-i} + \varepsilon_{t,(s_t)} + \sum_{j=1}^{m} \varphi_{j,(s_t)} \varepsilon_{t-j,(s_t)}, \\ \sigma^2_{t,(s_t)} &= w_{(s_t)} + \sum_{i=1}^{p} \alpha_{i,(s_t)} \varepsilon^2{}_{t-i,(s_t)} + \sum_{j=1}^{q} \beta_{(s_t)} \sigma_{t-j,(s_t)}, \end{aligned} \tag{2}$$

where,

$$\begin{aligned} \varepsilon_{t-i-1,(s_{t-i})} &= E\left[\varepsilon_{t-i-1,(s_{t-i-1})} \mid s_{t-i}, Y_{t-i-1}\right], \\ \sigma_{t-i-1,(s_{t-i})} &= E\left[\varepsilon_{t-i-1,(s_{t-i-1})} \mid s_{t-i}, Y_{t-i-1}\right]. \end{aligned} \tag{3}$$

Thus, the parameters have nonnegativity constraints $\phi, \theta, \varphi, w, \alpha, \beta > 0$ and the regimes are determined by $s_t$,

$$L = \prod_{t=1}^{T} f\left(y_t \mid s_t = i, Y_{t-1}\right) \Pr\left[s_t = i \mid Y_{t-1}\right], \tag{4}$$

and the probability $\Pr[s_t = i \mid Y_{t-1}]$ is calculated through iteration:

$$\begin{aligned} \pi_{jt} &= \Pr\left[s_t = j \mid Y_{t-1}\right] \\ &= \sum_{i=0}^{1} \Pr\left[s_t = j \mid s_{t-1} = i\right] \Pr\left[s_t = j \mid Y_{t-1}\right] \sum_{i=0}^{1} \eta_{ji} \pi^*_{it-1}. \end{aligned} \tag{5}$$

Accordingly, the two models, the Henneke et al. [108] and the Francq et al. [78] approaches, could be easily differentiated through the definitions of $\varepsilon^2_{t-1}$ and $\sigma_{t-1}$. Further, asymmetric power terms and fractional integration will be introduced to the derived model in the following sections.

### *2.2. MS-ARMA-APGARCH Model.*

Liu [99] provided a generalization of the Markov switching GARCH model of Haas et al. [81–83] and derived the conditions for stationarity and for the existence of moments. Liu [99] proposes a model which allowed for a nonlinear relation between past shocks and future volatility as well as for the leverage effects. The leverage effect is an outcome of the observation that the reaction of stock market volatility differed significantly to the positive and the negative innovations. Haas et al. [109, 110] complements Liu's [99] work in two ways. Firstly, the representation of the model developed by Haas [109] allows computational ease for obtaining the unconditional moments. Secondly, the dynamic autocorrelation structure of the power-transformed absolute returns (residuals) was taken as a measure of volatility.

Haas [109] model assumes that time series $\{\varepsilon_t, t \in \mathbf{Z}\}$ follows a $k$ regime MS-APGARCH process,

$$\varepsilon_t = \eta_t \sigma_{\Delta_t, t} \quad t \in \mathbf{Z}, \tag{6}$$

with $\{\eta_t, t \in \mathbf{Z}\}$ being *i.i.d.* sequence and $\{\Delta_t, t \in \mathbf{Z}\}$ is a Markov chain with finite state space $S = \{1, \ldots, k\}$ and $P$ is the irreducible and aperiodic transition matrix with typical element $p_{ij} = p(\Delta_t = j \mid \Delta_{t-1} = i)$ so that

$$P = \left[p_{ij}\right] = \left[p\left(\Delta_t = j \mid \Delta_{t-1} = i\right)\right], \quad i, j = 1, \ldots, k. \tag{7}$$

The stationary distribution of Markov-chain is shown as $\pi_\infty = (\pi_{1,\infty}, \ldots, \pi_{k,\infty})'$.

According to the Liu [99] notation of MS-APGARCH model, the conditional variance $\sigma^2_{jt}$ of $j$th regime follows a univariate APGARCH process as follows:

$$\sigma^\delta_{jt} = w_j + \alpha_{1j} \left|\varepsilon^+_{t-1}\right|^\delta + \alpha_{2j} \left|\varepsilon^-_{t-1}\right|^\delta + \beta_j \sigma^\delta_{j,t-1}, \quad \delta > 0, \tag{8}$$

where, $w_j > 0$, $\alpha_{1j}$, $\alpha_{2j}$, $\beta_j \geq 0$, $j = 1, \ldots, k$. For the power term $\delta = 2$ and for $\alpha_{1j} = \alpha_{2j}$, the model in (8) reduces to MS-GARCH model. Similar to the Ding et al. [19], the asymmetry, which is called "leverage effect," is captured by $\alpha_{1j} \neq \alpha_{2j}$ [109]. If the past negative shocks have deeper impact, parameters are expected to be $\alpha_{1j} < \alpha_{2j}$ so that the leverage effect becomes stronger.

Another approach that is similar to Liu [99] model is the Haas [109] model, where the asymmetry terms have a differentiated form as

$$\sigma_{jt}^{\delta} = w_j + \alpha_j\left(|\varepsilon_{t-1}| - \gamma_j\varepsilon_{t-1}\right)^{\delta} + \beta_j\sigma_{j,t-1}^{\delta}, \quad \delta > 0, \tag{9}$$

with the restrictions $0 < w_j$, $\alpha_j$, $\beta_j \geq 0$, $\gamma_j \in [-1, 1]$ with regimes $j = 1, \ldots, k$. The MS-APGARCH model of Haas [109] reduces to Ding et al. [19] single regime APGARCH model if $j = 1$. Equation (9) reduces to Liu [99] MS-APGARCH specification if $\alpha_{1j} = \alpha_j(1 - \gamma_j)^{\delta}$ and $\beta_j = \alpha_j^s(1 + \gamma_j)^{\delta}$.

The MS-ARMA-GARCH type model specification in this study assumes that the conditional mean follows MS-ARMA process, whereas the conditional variance follows regime switching in the GARCH architecture. Accordingly, MS-ARMA-APGARCH architecture nests several models by applying certain restrictions. The MS-ARMA-APGARCH model is derived by moving from MS-ARMA process in the conditional mean and MS-APGARCH$(l, m)$ conditional variance process as follows:

$$\begin{aligned}\sigma_{t,(s_t)}^{\delta_{(s_t)}} &= w_{(s_t)} + \sum_{l=1}^{r} \alpha_{l,(s_t)}\left(|\varepsilon_{t-l}| - \gamma_{l,(s_t)}\varepsilon_{t-l}\right)^{\delta_{(s_t)}} \\ &\quad + \sum_{m=1}^{q} \beta_{m,(s_t)}\sigma_{t-m,(s_t)}^{\delta_{(s_t)}}, \quad \delta_{(s_t)} > 0,\end{aligned} \tag{10}$$

where the regime switches are governed by $(s_t)$ and the parameters are restricted as $w_{(s_t)} > 0$, $\alpha_{l,(s_t)}$, $\beta_{m,(s_t)} \geq 0$ with $\gamma_{l,(s_t)} \in (-1, 1)$, $l = 1, \ldots, r$. One important difference is that MS-ARMA-APGARCH model in (10) allows the power parameters to vary across regimes. Further, if the following restrictions are applied, $l = 1$, $j = 1$, $\delta_{(s_t)} = \delta$, the model reduces to the model of Haas [109] given in (9).

In applied economics literature, it is shown that many financial time series possess long memory, which can be fractionally integrated. Fractional integration will be introduced to the MS-ARMA-APGARCH model given above.

*2.3. MS-ARMA-FIAPGARCH Model .* Andersen and Bollerslev [111], Baillie et al. [17], Tse [112], and Ding et al. [19] provided interesting applications in which the attention had been directed on long memory. Long memory could be incorporated to the model above by introducing fractional integration in the conditional mean and the conditional variance processes.

MS-ARMA-FIAPGARCH derived is a fractional integration augmented model as follows:

$$\begin{aligned}&\left(1 - \beta_{(s_t)}L\right)\sigma_{t,(s_t)}^{\delta_{(s_t)}} \\ &\quad = \omega + \left(\left(1 - \beta_{(s_t)}L\right) - \left(1 - \alpha_{(s_t)}L\right)(1 - L)^{d_{(s_t)}}\right) \\ &\qquad \times \left(|\varepsilon_{t-1}| - \gamma_{(s_t)}\varepsilon_{t-1}\right)^{\delta_{(s_t)}},\end{aligned} \tag{11}$$

where the lag operator is denoted by $L$, autoregressive parameters are $\beta_{(s_t)}$, and $\alpha_{(s_t)}$ shows the moving average parameters, $\delta_{(1)} > 0$ denotes the optimal power transformation, the fractional differentiation parameter varies between $0 \leq d_{(s_t)} \leq 1$ and allows long memory to be integrated to the model. Regime states $(s_t)$ are defined with $m$ regimes as $i = 1, \ldots, m$. The asymmetry term $|\gamma_{(s_t)}| < 1$ ensures that positive and negative innovations of the same size may have asymmetric effects on the conditional variance in different regimes.

Similar to the MS-ARMA-APGARCH model, the MS-ARMA-FIAPGARCH model nests several models. By applying $\delta_{(s_t)} = 2$, the model reduces to Markov switching fractionally integrated asymmetric GARCH (MS-ARMA-FIAGARCH); if $\delta_{(s_t)} = 2$ restriction is applied with $\gamma_{(s_t)} = 0$, the model reduces to Markov switching FIGARCH (MS-ARMA-FIGARCH). For $d_{(s_t)} = 0$, model reduces to the short memory version, the MS-ARMA-APGARCH model, if the additional constraint $\delta_{(s_t)} = 2$ is applied, the model reduces to MS-Asymmetric GARCH (MS-AGARCH). Lastly, for all the models mentioned above if $i = 1$, all models reduce to single regime versions of the relevant models, namely, the FIAPGARCH, FIAGARCH, FIGARCH, and AGARCH models, their relevant single regime variants. For a typical, with the constraints $i = 1$ and $\delta_{(s_t)} = 2$, $\gamma_{(s_t)} = 0$, the model reduces to single regime FIGARCH model of Baillie et al. [17]. To differentiate between the GARCH specifications, forecast performance criteria comparisons are assumed.

## 3. Neural Network and MS-ARMA-GARCH Models

In this section of the study, the MultiLayer Perceptron, Radical Basis Function, and Recurrent Neural Network models that belong to the ANN family will be combined with Markov switching and GARCH models. In this respect, Spezia and Paroli [113] is another study that merged the Neural Network and MS-ARCH models.

### *3.1. Multilayer Perceptron (MLP) Models*

*3.1.1. MS-ARMA-GARCH-MLP Model.* Artificial Neural Network models have many applications in modeling of functional forms in various fields. In economics literature, the early studies such as Dutta and Shektar [114], Tom and Kiang [115], Do and Grudinsky [116], Freisleben [51], and Refenes et al. [55] utilize ANN models to option pricing, real estates, bond ratings, and prediction of banking failures among many, whereas studies such as Kanas [40], Kanas and

Yannopoulos [65], and Shively [117] applied ANN models to stock return forecasting, and Donaldson and Kamstra [118] proposed hybrid modeling to combine GARCH, GJR, and EGARCH models with ANN architecture.

The MLP, an important class of neural networks, consists of a set of sensory units that constitute the input layer, one or more hidden layers, and an output layer. The additional linear input which is connected to the MLP network is called the Hybrid MLP. Hamilton model can also be considered as a nonlinear mixture of autoregressive functions, such as the multilayer perceptron and thus, the Hamilton model is called Hybrid MLP-HMC models [119]. Accordingly, in the HMC model, the regime changes are dominated by a Markov chain without making a priori assumptions in light of the number of regimes [119]. In fact, Hybrid MLP accepts the network inputs to be connected to the output nodes with weighted connections to form a linear model that is parallel with nonlinear Multilayer Perceptron.

In the study, the MS-ARMA-GARCH-MLP model to be proposed allows Markov switching type regime changes both in the conditional mean and conditional variance processes augmented with MLP type neural networks to achieve improvement in terms of in-sample and out-of-sample forecast accuracy.

The MS-ARMA-GARCH-MLP model is defined of the form:

$$y_t = c_{(s_t)} + \sum_{i=1}^{r} \theta_{i,(s_t)} y_{t-i} + \varepsilon_{t,(s_t)} + \sum_{j=1}^{n} \varphi_{j,(s_t)} \varepsilon_{t-j,(s_t)}, \tag{12}$$

$$\begin{aligned}\sigma^2_{t,(s_t)} = {} & w_{(s_t)} + \sum_{p=1}^{p} \alpha_{p,(s_t)} \varepsilon^2_{t-p,(s_t)} + \sum_{q=1}^{q} \beta_{q,(s_t)} \sigma_{t-q,(s_t)} \\ & + \sum_{h=1}^{h} \xi_{h,(s_t)} \psi\left(\tau_{h,(s_t)}, Z_{t,(s_t)} \lambda_{h,(s_t)}, \theta_{h,(s_t)}\right),\end{aligned} \tag{13}$$

where, the regimes are governed by unobservable Markov process:

$$\sum_{i=1}^{m} \sigma^2_{t(i)} P\left(S_t = i \mid z_{t-1}\right), \quad i = 1, \ldots m. \tag{14}$$

In the MLP type neural network, the logistic type sigmoid function is defined as

$$\begin{aligned}&\psi\left(\tau_{h,(s_t)}, Z_{t,(s_t)} \lambda_{h,(s_t)}, \theta_{h,(s_t)}\right) \\ &= \left[1 + \exp\left(-\tau_{h,(s_t)} \left(\sum_{l=1}^{l} \left[\sum_{h=1}^{h} \lambda_{h,l,(s_t)} z^h_{t-l,(s_t)} + \theta_{h,(s_t)}\right]\right)\right)\right]^{-1}\end{aligned} \tag{15}$$

$$\left(\frac{1}{2}\right) \lambda_{h,d} \sim \text{ uniform } [-1, +1] \tag{16}$$

and $P(S_t = i \mid z_{t-1})$, the filtered probability with the following representation,

$$\left(P\left(S_t = i \mid z_{t-1}\right) \alpha f\left(P\left(\sigma_{t-1} \mid z_{t-1}, s_{t-1} = 1\right)\right)\right) \tag{17}$$

if $n_{j,i}$ transition probability $P(s_t = i \mid s_{t-1} = j)$ is accepted;

$$z_{t-d} = \frac{\left[\varepsilon_{t-d} - E(\varepsilon)\right]}{\sqrt{E\left(\varepsilon^2\right)}} \tag{18}$$

$s \rightarrow \max\{p, q\}$ recursive procedure is started by constructing $P(z_s = i \mid z_{s-1})$, where $\psi(z_t \lambda_h)$ is of the form $1/(1 + \exp(-x))$, a twice-differentiable, continuous function bounded between $[0, 1]$. The weight vector $\xi = w$; $\psi = g$ logistic activation function and input variables are defined as $z_t \lambda_h = x_i$, where $\lambda_h$ is defined as in (16).

If $n_{j,i}$ transition probability $P(z_t = i \mid z_{t-1} = j)$ is accepted,

$$\begin{aligned}&f\left(y_t \mid x_t, z_t = i\right) \\ &= \frac{1}{\sqrt{2\pi h_{t(i)}}} \exp\left\{\frac{-\left(y_t - x_t' \varphi - \sum_{j=1}^{H} \beta_j p\left(x_t' \gamma_j\right)\right)^2}{2 h_{t(j)}}\right\},\end{aligned} \tag{19}$$

$s \rightarrow \max\{p, q\}$, recursive procedure is started by constructing $P(z_s = i \mid z_{s-1})$.

*3.1.2. MS-ARMA-APGARCH-MLP Model.* Asymmetric power GARCH (APGARCH) model has interesting features. In the construction of the model, the APGARCH structure of Ding et al. [19] is followed. The model given in (13) is modified to obtain the Markov switching APGARCH Multilayer Perceptron (MS-ARMA-APGARCH-MLP) model of the form,

$$\begin{aligned}&\sigma^{\delta,(s_t)}_{t,(s_t)} \\ &= w_{(s_t)} + \sum_{p=1}^{p} \alpha_{p,(s_t)} \left(\left|\varepsilon_{t-p}\right| - \gamma_{p,(s_t)} \varepsilon_{t-p,(s_t)}\right)^{\delta,(s_t)} \\ &\quad + \sum_{q=1}^{q} \beta_{q,(s_t)} \sigma^{\delta,(s_t)}_{t-q,(s_t)} + \sum_{h=1}^{h} \xi_{h,(s_t)} \psi\left(\tau_{h,(s_t)}, Z_{t,(s_t)} \lambda_{h,(s_t)}, \theta_{h,(s_t)}\right),\end{aligned} \tag{20}$$

where, regimes are governed by unobservable Markov process. The model is closed as defining the conditional mean as in (12) and conditional variance of the form equation's (14)–(19) and (20) to augment the MS-ARMA-GARCH-MLP model with asymmetric power terms to obtain MS-ARMA-APGARCH-MLP. Note that, model nest several specifications. Equation (20) reduces to the MS-ARMA-GARCH-MLP model in (13) if the power term $\delta = 2$ and $\gamma_{p,(s_t)} = 0$. Similarly, the model nests MS-GJR-MLP if $\delta = 2$ and $0 \leq \gamma_{p,(s_t)} \leq 1$ are imposed. The model may be shown as MSTGARCH-MLP model if $\delta = 1$ and $0 \leq \gamma_{p,(s_t)} \leq 1$. Similarly, by applying a single regime restriction, $s_t = s = 1$, the quoted models reduce to their respective single regime variants, namely, the ARMA-APGARCH-MLP, ARMA-GARCH-MLP, ARMA-NGARCH-MLP, ARMA-GJRGARCH-MLP, and ARMA-GARCH-MLP models (for further discussion in NN-GARCH family models, see Bildirici and Ersin [61]).

*3.1.3. MS-ARMA-FIAPGARCH-MLP Model.* Following the methodology discussed in the previous section, MS-ARMA-APGARCH-MLP model is augmented with neural network modeling architecture and that accounts for fractional integration to achieve long memory characteristics to obtain MS-ARMA-FIAPGARCH-MLP. Following the MS-ARMA-FIAPGARCH represented in (11), the MLP type neural network augmented MS-ARMA-FIAPGARCH-MLP model representation is achieved:

$$\begin{aligned}
&\left(1-\beta_{(s_t)}L\right)\sigma_{t,(s_t)}^{\delta_{(s_t)}} \\
&\quad = w_{(s_t)} + \left(\left(1-\beta_{(s_t)}L\right) - \left(1-\phi_{(s_t)}L\right)(1-L)^{d_{(s_t)}}\right) \\
&\quad\quad \times \left(\left|\varepsilon_{t-1,(s_t)}\right| - \gamma_{(s_t)}\varepsilon_{t-1,(s_t)}\right)^{\delta_{(s_t)}} \\
&\quad\quad + \sum_{h=1}^{h}\xi_{h,(s_t)}\psi\left(\tau_{h,(s_t)}, Z_{t,(s_t)}\lambda_{h,(s_t)}, \theta_{h,(s_t)}\right),
\end{aligned} \tag{21}$$

where, $h$ are neurons defined with sigmoid type logistic functions, $i = 1, \ldots, m$ regime states governed by unobservable variable following Markov process. Equation (21) defines the MS-ARMA-FIAPGARCH-MLP model, the fractionally integration variant of the MSAGARCH-MLP model modified with the ANN, and the logistic activation function, $\psi(\tau_{h,(s_t)}, Z_{t,(s_t)}\lambda_{h,(s_t)}, \theta_{h,(s_t)})$ defined as in (15). Bildirici and Ersin [61] proposes a class of NN-GARCH models including the NN-APGARCH. Similarly, the MS-ARMA-FIAPGARCH-MLP model reduces to the MS-FIGARCH-MLP model for restrictions on the power term $\delta_{(s_t)} = 2$ and $\gamma_{(s_t)} = 0$. Further, the model reduces to MS-FINGARCH-MLP model for $\gamma_{(s_t)} = 0$ and to the MS-FI-GJRGARCH-MLP model if $\delta_{(s_t)} = 2$ and $\gamma_{(s_t)}$ is restricted to be in the range of $0 \leq \gamma_{(s_t)} \leq 1$. The model reduces to MS-TGARCH-MLP model if $\delta_{(s_t)} = 1$ in addition to the $0 \leq \gamma_{(s_t)} \leq 1$ restriction. On the contrary, if single regime restriction is imposed, models discussed above, namely, MS-ARMA-FIAPGARCH-MLP, MSFIGARCH-NN, MSFIGARCH-NN, MSFINGARCH-MLP, MSFIGJRGARCH-MLP, and MSFITGARCH-MLP models reduce to NN-FIAPGARCH, NN-FIGARCH, NN-FIGARCH, NN-FINGARCH, NN-FIGJRGARCH, and NN-FITGARCH models, which are single regime neural network augmented GARCH family models of the form Bildirici and Ersin [61] that do not possess Markov switching type asymmetry (Bildirici and Ersin [61]). The model also nests model variants that do not possess long memory characteristics. By imposing $d_{(s_t)} = 0$ to the fractional integration parameter which may take different values under $i = 1, 2, \ldots, m$ different regimes, the model in (21) reduces to MS-ARMA-APGARCH-MLP model, the short memory model variant. In addition to the restrictions applied above, application of $d_{(s_t)} = 0$ results in models without long memory characteristics: MS-ARMA-FIAPGARCH-MLP, MS-ARMA-GARCH-MLP, MS-ARMA-GARCH-MLP, MSNGARCH-MLP, MS-GJR-GARCH-MLP, and MSTGARCH-MLP.

For a typical example, consider a MS-ARMA-FIAPGARCH-MLP model representation with two regimes:

$$\begin{aligned}
&\left(1-\beta_{(1)}L\right)\sigma_{t,(1)}^{\delta_{(1)}} \\
&\quad = w_{(1)} + \left(\left(1-\beta_{(1)}L\right) - \left(1-\phi_{(1)}L\right)(1-L)^{d_{(1)}}\right) \\
&\quad\quad \times \left(\left|\varepsilon_{t-1}\right| - \gamma_{(1)}\varepsilon_{t-1}\right)^{\delta_{(1)}} + \sum_{h=1}^{h}\xi_{h,(1)}\psi\left(\tau_{h,(1)}, Z_{t,(1)}\lambda_{h,(1)}, \theta_{h,(1)}\right), \\
&\left(1-\beta_{(2)}L\right)\sigma_{t,(2)}^{\delta_{(2)}} \\
&\quad = w_{(2)} + \left(\left(1-\beta_{(2)}L\right) - \left(1-\phi_{(2)}L\right)(1-L)^{d_{(2)}}\right) \\
&\quad\quad \times \left(\left|\varepsilon_{t-1}\right| - \gamma_{(2)}\varepsilon_{t-1}\right)^{\delta_{(2)}} + \sum_{h=1}^{h}\xi_{h,(2)}\psi\left(\tau_{h,(2)}, Z_{t,(2)}\lambda_{h,(2)}, \theta_{h,(2)}\right).
\end{aligned} \tag{22}$$

Following the division of regression space into two subspaces with Markov switching, the model allows two different asymmetric power terms, $\delta_{(1)}$ and $\delta_{(2)}$, and two different fractional differentiation parameters, $d_{(1)}$ and $d_{(2)}$; as a result, different long memory and asymmetric power structures are allowed in two distinguished regimes.

It is possible to show the model as a single regime NN-FIAPGARCH model if $i = 1$:

$$\begin{aligned}
&(1-\beta L)\sigma_t^{\delta} \\
&\quad = \omega + \left((1-\beta L) - (1-\phi L)(1-L)^{d}\right) \\
&\quad\quad \times \left(\left|\varepsilon_{t-1}\right| - \gamma_j\varepsilon_{t-1}\right)^{\delta} + \sum_{h=1}^{h}\xi_h\psi\left(\tau_h, Z_t\lambda_h, \theta_h\right).
\end{aligned} \tag{23}$$

Further, the model reduces to NN-FIGARCH if $i = 1$ and $\delta_{(s_1)} = \delta = 2$ in the fashion of Bildirici and Ersin [61]:

$$\begin{aligned}
&(1-\beta L)\sigma_t^{2} \\
&\quad = \omega + \left((1-\beta L) - (1-\phi L)(1-L)^{d}\right) \\
&\quad\quad \times \left(\left|\varepsilon_{t-1}\right| - \gamma_j\varepsilon_{t-1}\right)^{2} + \sum_{h=1}^{h}\xi_h\psi\left(\tau_h, Z_t\lambda_h, \theta_h\right).
\end{aligned} \tag{24}$$

*3.2. Radial Basis Function Model.* *Radial Basis Functions* are one of the most commonly applied neural network models that aim at solving the interpolation problem encountered in nonlinear curve fitting.Liu and Zhang [120] utilized the *Radial Basis Function Neural Networks (RBF)* and Markov regime-switching regressionsto divide the regression space into two sub-spaces to overcome the difficulty in estimating the conditional volatility inherent in stock returns. Further, Santos et al. [121] developed a RBF-NN-GARCH model that benefit from the RBF type neural networks. Liu and Zhang [120] combined RBF neural network models with the Markov Switching model to merge Markov switching Neural Network model based on RBF models. RBF neural networks in their models are trained to generate both time series forecasts and certainty factors. Accordingly, RBF neural network is represented as a composition of three layers of nodes; first, the input layer that feeds the input data to each of the nodes in the second or hidden layer; the second layer that differs from other neural networks in that each node represents a data cluster which is centered at a particular point and has a given radius and in the third layer, consisting of one node.

*3.2.1. MS-ARMA-GARCH-RBF Model.* MS-GARCH-RBF model is defined as

$$\begin{aligned}\sigma^2_{t,(s_t)} &= w_{(s_t)} + \sum_{p=1}^{P} \alpha_{P,(s_t)} \varepsilon^2_{t-P,(s_t)} \\ &\quad + \sum_{q=1}^{q} \beta_{q,(s_t)} \sigma_{t-q,(s_t)} \\ &\quad + \sum_{h=1}^{h} \xi_{h,(s_t)} \phi_{h,(s_t)} \left( \left\| Z_{t,(s_t)} - \mu_{h,(s_t)} \right\| \right),\end{aligned} \tag{25}$$

where $i = 1, \ldots, m$ regimes are governed by unobservable Markov process:

$$\sum_{i=1}^{m} \sigma^2_{t,(s_t)} P\left(S_t = i \mid z_{t-1}\right). \tag{26}$$

A Gaussian basis function for the hidden units given as $\phi(x)$ for $x = 1, 2, \ldots, X$ where the activation function is defined as,

$$\phi\left(h, (s_t), Z_t\right) = \exp\left( \frac{-\left\| Z_{t,(s_t)} - \mu_{h,(s_t)} \right\|^2}{2\rho^2} \right). \tag{27}$$

With $p$ defining the width of each function. $Z_t$ is a vector of lagged explanatory variables, $\alpha + \beta < 1$ is essential to ensure stationarity. Networks of this type can generate any real-valued output, but in their applications where they have a priori knowledge of the range of the desired outputs, it is computationally more efficient to apply some nonlinear transfer function to the outputs to reflect that knowledge.

$P(S_t = i \mid z_{t-1})$ is the filtered probability with the following representation:

$$\left(P\left(S_t = i \mid z_{t-1}\right) \alpha f\left(P\left(\sigma_{t-1} \mid z_{t-1}, s_{t-1} = 1\right)\right)\right). \tag{28}$$

If $n_{j,i}$ transition probability $P(s_t = i \mid s_{t-1} = j)$ is accepted,

$$z_{t-d} = \frac{\left[\varepsilon_{t-d} - E(\varepsilon)\right]}{\sqrt{E\left(\varepsilon^2\right)}} \tag{29}$$

$s \rightarrow \max\{p, q\}$ recursive procedure is started by constructing $P(z_s = i \mid z_{s-1})$.

*3.2.2. MS-ARMA-APGARCH-RBF Model.* Radial basis functions are three-layer neural network models with linear output functions and nonlinear activation functions defined as Gaussian functions in hidden layer utilized to the inputs in light of modeling a radial function of the distance between the inputs and calculated value in the hidden unit. The output unit produces a linear combination of the basis functions to provide a mapping between the input and output vectors:

$$\begin{aligned}\sigma^{\delta,(s_t)}_{t,(s_t)} &= w_{(s_t)} + \sum_{p=1}^{p} \alpha_{p,(s_t)} \left( \left| \varepsilon_{t-j} \right| - \gamma_{p,(s_t)} \varepsilon_{t-p,(s_t)} \right)^{\delta,(s_t)} \\ &\quad + \sum_{q=1}^{q} \beta_{q,(s_t)} \sigma^{\delta,(s_t)}_{t-q,(s_t)} \\ &\quad + \sum_{h=1}^{h} \xi_{h,(s_t)} \phi_{h,(s_t)} \left( \left\| Z_{t,(s_t)} - \mu_{h,(s_t)} \right\| \right),\end{aligned} \tag{30}$$

where, $i = 1, \ldots, m$ regime model and regimes are governed by unobservable Markov process. Equations (26)–(29) with (30) define the MS-ARMA-APGARCH-RBF model. Similar to the MS-ARMA-APGARCH-MLP model, the MS-ARMA-APGARCH-RBF model nests several models. Equation (30) reduces to the MS-ARMA-GARCH-RBF model if the power term $\delta = 2$ and $\gamma_{p,(s_t)} = 0$, to the MSGARCH-RBF model for $\gamma_{p,(s_t)} = 0$, and to the MSGJRGARCH-RBF model if $\delta = 2$ and $0 \le \gamma_{p,(s_t)} \le 1$ restrictions are allowed. The model may be shown as MSTGARCH-RBF model if $\delta = 1$ and $0 \le \gamma_{p,(s_t)} \le 1$. Further, single regime models, namely, NN-APGARCH, NN-GARCH, NN-GARCH, NN-NGARCH, NN-GJRGARCH, and NN-TGARCH models, may be obtained if $t = 1$ (for further discussion in NN-GARCH family models, see Bildirici and Ersin [61]).

*3.2.3. MS-ARMA-FIAPGARCH-RBF Model.* MS-FIAPGARCH-RBF model is defined as

$$\begin{aligned}&\left(1 - \beta_{(s_t)} L\right) \sigma^{\delta_{(s_t)}}_{t,(s_t)} \\ &\quad = w_{(s_t)} + \left( \left(1 - \beta_{(s_t)} L\right) - \left(1 - \phi_{(s_t)} L\right) (1 - L)^{d_{(s_t)}} \right) \\ &\qquad \times \left( \left| \varepsilon_{t-1,(s_t)} \right| - \gamma_{(s_t)} \varepsilon_{t-1,(s_t)} \right)^{\delta_{(s_t)}} \\ &\qquad + \sum_{h=1}^{h} \xi_{h,(s_t)} \phi_{h,(s_t)} \left( \left\| Z_{t,(s_t)} - \mu_{h,(s_t)} \right\| \right),\end{aligned} \tag{31}$$

where, $h$ are neurons defined with Gaussian functions. The MS-ARMA-FIAPGARCH-RBF model is a variant of the MSAGARCH-RBF model with fractional integration augmented with ANN architecture. Similarly, the MS-ARMA-FIAPGARCH-RBF model reduces to the MS-ARMA-FIGARCH-RBF model with restrictions on the power term $\delta_{(s_t)} = 2$ and $\gamma_{(s_t)} = 0$. The model nests MSFINGARCH-RBF model for $\gamma_{(s_t)} = 0$, and MSFIGJRGARCH-RBF model if $\delta_{(s_t)} = 2$ and $\gamma_{(s_t)}$ varies between $0 \le \gamma_{(s_t)} \le 1$. Further, the model may be shown as MSTGARCH-RBF model if $\delta_{(s_t)} = 1$ and $0 \le \gamma_{(s_t)} \le 1$. With single regime restriction $i = 1$, discussed models reduce to NN-FIAPGARCH, NN-FIGARCH, NN-FIGARCH, NN-FINGARCH, NN-FIGJRGARCH, and NN-FITGARCH models, which do not possess Markov switching type asymmetry. To obtain the model with short memory characteristics, $d_{(\cdot)} = 0$ restriction on fractional integration parameters should be imposed and the model reduces to MSAPGARCH-RBF model, the short memory model variant. Additionally, by applying $d_{(\cdot)} = 0$ with the restrictions discussed above, models without long memory characteristics: MSFIAPGARCH-RBF, MSGARCH-RBF, MSGARCH-RBF, MSNGARCH-RBF, MSGJRGARCH-RBF, and MSTGARCH-RBF models could be obtained.

*3.3. Recurrent Neural Network MS-GARCH Models.* The RNN model includes the feed-forward system; however, it distinguishes itself from standard feed-forward network models in the activation characteristics within the layers. The activations are allowed to provide a feedback to units within the same or preceding layer(s). This forms an internal memory system that enables a RNN to construct sensitive internal representations in response to temporal features found within a data set.

The Jordan [122] and Elman's [123] networks are simple recurrent networks to obtain forecasts: Jordan and Elman networks extend the multilayer perceptron with context units, which are processing elements (PEs) that remember past activity. Context units provide the network with the ability to extract temporal information from the data. The RNN model employs back propagation-through-time, an efficient gradient-descent learning algorithm for recurrent networks. It was used as a standard variant of cross-validation referred to as the leave-one-out method and as a stopping criterion suitable for estimation problems with sparse data and so it is identified the onset of overfitting during training. The RNN was functionally equivalent to a nonlinear regression model used for time-series forecasting (Zhang et al. [124]; Binner et al. [125]). Tiňo et al. [126] merged the RNN and GARCH models.

*3.3.1. MS-ARMA-GARCH-RNN Models.* The model is defined as

$$\begin{aligned}\sigma^2_{t,(s_t)} &= w_{(s_t)} + \sum_{p=1}^{p} \alpha_{p,(s_t)} \varepsilon^2_{t-p,(s_t)} + \sum_{q=1}^{q} \beta_{q,(s_t)} \sigma_{t-q,(s_t)} \\ &\quad + \sum_{h=1}^{h} \xi_{h,(s_t)} \pi_{h,(s_t)} \left( w_{k,h,(s_t)} \theta_{t-k} + \theta_{k,h,(s_t)} \right).\end{aligned} \tag{32}$$

Similar to the models above, (32) is shown for $i = 1, \ldots, m$ regimes which are governed by unobservable Markov process. Activation function is taken as the logistic function.

*3.3.2. MS-ARMA-APGARCH-RNN.* Markov switching APGARCH Recurrent Neural Network Model is represented as

$$\begin{aligned}\sigma^{\delta,(s_t)}_{t,(s_t)} &= w_{(s_t)} + \sum_{p=1}^{p} \alpha_{p,(s_t)} \left( \left| \varepsilon_{t-j} \right| - \gamma_{p,(s_t)} \varepsilon_{t-p,(s_t)} \right)^{\delta,(s_t)} \\ &\quad + \sum_{q=1}^{q} \beta_{q,(s_t)} \sigma^{\delta,(s_t)}_{t-q,(s_t)} \\ &\quad + \sum_{h=1}^{h} \xi_{h,(s_t)} \Pi \left( \theta_{k,h,(s_t)} \chi_{t-k,h,(s_t)} + \theta_{k,h,(s_t)} \right)\end{aligned} \tag{33}$$

$i = 1, \ldots, m$ regimes are governed by unobservable Markov process. $\theta_{k,h,(s_t)}$ is the weights of connection from pre to postsynaptic nodes, $\Pi(x)$ is a logistic sigmoid function of the form given in (15), $\chi_{t-k,h,(s_t)}$ is a variable vector corresponding to the activations of postsynaptic nodes, the output vector of the hidden units, and $\theta_{k,h,(s_t)}$ are the bias parameters of the presynaptic nodes and $\xi_{i,(s_t)}$ are the weights of each hidden unit for $h$ hidden neurons, $i = 1, \ldots, h$. The parameters are estimated by minimizing the sum of the squared-error loss: $\min \lambda = \sum_{t-1}^{T} [\sigma_t - \hat{\sigma}_t]^2$. The model is estimated by recurrent back-propagation algorithm and by the recurrent Newton algorithm. By imposing several restrictions similar to the MS-ARMA-APGARCH-RBF model, several representations are shown under certain restrictions. Equation (33) reduces to MS-ARMA-GARCH-RNN model with $\delta = 2$ and $\gamma_{p,(s_t)} = 0$, to the MSGARCH-RNN model for $\gamma_{p,(s_t)} = 0$, and to the MSGJRGARCH-RNN model if $\delta = 2$ and $0 \leq \gamma_{p,(s_t)} \leq 1$ restrictions are imposed. MSTGARCH-RNN model is obtained if $\delta = 1$ and $0 \leq \gamma_{p,(s_t)} \leq 1$. In addition to the restrictions above, if the single regime restriction $i = 1$ is implied, the model given in Equation (33) reduces to their single regime variants; namely, the APGARCH-RNN, GARCH-RNN, GJRGARCH-RNN, and TGARCH-RNN models, respectively.

*3.3.3. MS-ARMA-FIAPGARCH-RNN.* Markov Switching Fractionally Integrated APGARCH Recurrent Neural Network Model is defined as

$$\begin{aligned}&\left(1 - \beta_{(s_t)} L\right) \sigma^{\delta_{(s_t)}}_{t,(s_t)} \\ &\quad = w_{(s_t)} + \left( \left(1 - \beta_{(s_t)} L\right) - \left(1 - \phi_{(s_t)} L\right) (1 - L)^{d_{(s_t)}} \right) \\ &\qquad \times \left( \left| \varepsilon_{t-1,(s_t)} \right| - \gamma_{(s_t)} \varepsilon_{t-1,(s_t)} \right)^{\delta_{(s_t)}} \\ &\qquad + \sum_{h=1}^{h} \xi_{h,(s_t)} \Pi \left( \theta_{k,h,(s_t)} \chi_{t-k,h,(s_t)} + \theta_{k,h,(s_t)} \right),\end{aligned} \tag{34}$$

where, $h$ are neurons defined as sigmoid type logistic functions and $i = 1, \ldots, m$ regime states the following Markov process. The MS-ARMA-FIAPGARCH-RNN model is the fractionally integrated variant of the MS-ARMA-APGARCH-RNN model. The MS-ARMA-FIAPGARCH-RNN model reduces to the MS-ARMA-FIGARCH-RNN model with restrictions on the power term $\delta_{(s_t)} = 2$ and $\gamma_{(s_t)} = 0$. Further, the model reduces to MSFINGARCH-RNN model for $\gamma_{(s_t)} = 0$, to the MSFIGJRGARCH-RNN model if $\delta_{(s_t)} = 2$

and $0 \leq \gamma_{(s_t)} \leq 1$. The model reduces to MSFITGARCH-RNN model for $0 \leq \gamma_{(s_t)} \leq 1$ and $\delta_{(s_t)} = 1$. Single regime restriction $i = 1$ leads to the NN-FIAPGARCH-RNN, NN-FIGARCH-RNN, NN-FIGARCH-RNN, NN-FINGARCH-RNN, NN-FIGJRGARCH-RNN, and NN-FITGARCH-RNN models without Markov switching.

## 4. Data and Econometric Results

*4.1. The Data.* In order to test forecasting performance of the abovementioned models, stock return in Turkey is calculated by using the daily closing prices of Istanbul Stock Index ISE 100 covering the 07.12.1986–13.12.2010 period corresponding to 5852 observations. To obtain return series, the data is calculated as follows: $y_t = \ln(P_t/P_{t-1})$, where $\ln(\cdot)$ is the natural logarithms, $P_t$ is the ISE 100 index, and $y_t$ is taken as a measure of stock returns. In the process of training the models, the sample is divided between training, test, and out-of-sample subsamples with the percentages of 80%, 10%, and 10%. Further, we took the sample size for the training sample as the first 4680 observations, whereas the sample sizes for the test and out-of-sample samples are 585 and 587. The statistics of daily returns calculated from ISE 100 Index are given in Table 1.

In order to provide out-of-sample forecasts of the ISE100 daily returns, two competing nonlinear model structures are used, the univariate Markov switching model and Neural Network Models, MLP, RBF, and RNN. In order to assess the predictability of models, models are compared for their out-of-sample forecasting performance. Firstly, by calculating RMSE and MSE error criteria, the forecast comparisons are obtained.

*4.2. Econometric Results.* At the first stage, selected GARCH family models taken as baseline models are estimated for evaluation purposes. Results are given in Table 2. Included models have different characteristics to be evaluated, namely, fractional integration, asymmetric power, and fractionally integrated asymmetric power models, namely, GARCH, APGARCH, FIGARCH, and FIAPGARCH models. Random walk (RW) model is estimated for comparison purpose. Furthermore, the models given in Table 2 will provide basis for nonlinear models to be estimated.

It is observed that, though all volatility models perform better than the RW model in light of Log Likelihood criteria, as we move from the GARCH model to asymmetric power GARCH (APGARCH) model, the fit of the models improve accordingly. The sum of ARCH and GARCH parameters is calculated as 0.987 and less than 1. The results for the APGARCH model show that the calculated power term is 1.35 and the asymmetry is present. Further, similar to the findings of McKenzie and Mitchell [127], it is observed that the addition of the leverage and power terms improves generalization power and thus show that squared power term may not necessarily be optimal as Ding et al. [19] study suggested.

In Table 3, transition matrix and the MS model were estimated. The standard deviation takes the values of 0.05287 and 0.014572 for regime 1 and regime 2. It lasts approximately 75.87 months in regime 1 and 107.61 months in regime 2. By using maximum likelihood approach, MS-GARCH models are tested by assuming that the error terms follow student-$t$ distribution with the help of BFGS algorithm. Number of regimes are taken as 2 and 3. GARCH effect in the residuals is tested and at 1% significance level, the hypothesis that there are no GARCH effects is rejected. Additionally, the normality in the residuals is tested with Jacque-Berra test, at 1% significance level, it is detected that the residuals are not normally distributed. As a result, MS-GARCH model is estimated under the $t$ distribution assumption. In the MS-GARCH model, the transition probability results are calculated as $\text{Prob}(s_t = 1 \mid s_{t-1} = 1) = 0.50$ and $\text{Prob}(s_t = 2 \mid s_{t-1} = 2) = 0.51$ and show that the persistence is low in the MS-GARCH model.

Statistical inference regarding the empirical validity of two-regime switching process was carried out by using nonstandard LR tests [128]. The nonstandard LR test is statistically significant and this suggests that linearity is strongly rejected.

The volatility values tend to be calculated at higher values than they actually are in GARCH models. As the persistence coefficients obtained for the MS-GARCH, MS-PGARCH, and MS-APGARCH models are compared to those obtained for the GARCH models,; the persistence in the GARCH models are comparatively higher and for the GARCH models; a shock in volatility is persistent and shows continuing effect. This situation occurs as a result of omitting the importance of structural change in the ARCH process. In the MS-GARCH, MS-PGARCH, and MS-APGARCH models which take this situation into consideration, the value of persistence parameter decreases.

Power terms are reported comparatively lower for developed countries than the less developed countries. Haas et al. [109, 110] calculated three state RS-GARCH, RS-PGARCH, and RS-APGARCH models for the daily returns in NYSE and estimated the power terms for the RS-APGARCH model as 1.25, 1.09, and 1.08. For Turkey, Ural [129] estimated RS-APGARCH models for returns in ISE100 index in Turkey in addition to United Kingdom FTSE100, CAC40 in France and NIKKEI 225 indices in Japan and reported highest power estimates (1.84) compared to the power terms calculated as 1.26, 1.31, and 1.24 for FTSE100, NIKKEI 225, and CAC40. Further, power terms obtained for returns calculated for stock indices in many developing economies are calculated comparatively higher than those obtained for the various indices in developed countries. Ané and Ureche-Rangau [130] estimated single regime GARCH and APGARCH models in addition to RS-GARCH and RS-APGARCH models following Gray [92] model. Power terms in single regime APGARCH models were calculated for daily returns as 1.57 in Nikkei 225 Index, as 1.81 in Hang Seng Index, as 1.69 in Kuala Lumpur Composite Index, and as 2.41 in Singapore SES-ALL Index, whereas, for regime switching APGARCH models, power terms are calculated as 1.20 in regime 1 and 1.83 in regime 2 for Nikkei 225 Index, 2.16 in regime 1 and 2.31 in regime 2 for Heng Seng Index, 1.95 and 2.17 for regimes 1 and 2 in Singapore SES-ALL Index, and 1.71 and 2.25 in

Table 1: Daily returns in ISE 100 index, basic statistics.

| Mean | Median | Mode | Minimum | Maximum | Standard deviation | Skewness | Kurtosis |
|---|---|---|---|---|---|---|---|
| 0.001725 | 0.001421 | 0.0000001 | −0.19979 | 0.217108 | 0.028795 | 0.071831 | 4.207437 |

Table 2: Baseline volatility models.

| Baseline GARCH models, single regime | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| (1) RW | C | | | | | | | log $L$ |
| | 0.001873$^{***}$<br>(0.000391) | | | | | | | 12241.56 |
| (2) GARCH | ARCH | GARCH | C | | | | | log $L$ |
| | 0.1591259$^{***}$<br>(0.0068625) | 0.8280324$^{***}$<br>(0.0056163) | 0.0000206$^{***}$<br>$(1.80e-06)$ | | | | | 13174.66 |
| (3) APGARCH | APARCH | APARCH_E | PGARCH | POWER | C | | | log $L$ |
| | 0.2153934$^{***}$<br>(0.0076925) | −0.0319876$^{***}$<br>(0.0160775) | 0.7801279$^{***}$<br>(0.0074049) | 1.354999$^{***}$<br>(0.0707303) | 0.0004055$^{***}$<br>(0.0001074) | | | 13111.96 |
| (4) FIAPGARCH | ARCH<br>(Phi1) | GARCH<br>(Beta1) | D-FIGARCH | APARCH<br>(Gamma1) | APARCH<br>(Delta) | C<br>(MEAN) | C<br>(VAR.) | log $L$ |
| | 0.250877$^{***}$<br>(0.096563) | 0.449631$^{***}$<br>(0.10367) | 0.417296$^{***}$<br>(0.040388) | 0.027382<br>(0.033565) | 2.020946$^{***}$<br>(0.096663) | 0.001635$^{***}$<br>(0.00028929) | 8.054283<br>(6.6685) | 13196.30 |

regimes 1 and 2 for the model calculated for returns in Kuala Lumpur Composite Index. Teletar and Binay [131] estimated APARCH models for Turkey and 10 national stock indices and noted that power terms reported for developing countries tend to be high and varying though those reported for the developed countries are estimated with low and close values. They estimated power terms for ISE100 index as 1.960 for 1987–2001 period and 1.48 for 1989–1996 period, whereas power terms for S&P, FTSE, NIKKEI, Hong Kong, NZSE40, DAX index of Germany, CAC40 of France, Singapore's SES, TSE, ALORAI, and MSCI indices as 1.21, 1.43, 1.22, 1.35, 1.37, 0.91, 1.17, 2.48, 1.45, 1.01, and 1.37, respectively.

On the other hand, though the improvement by shifting to modeling the conditional volatility with regime switching is noteworthy, the desired results are still not obtained, therefore, MS-GARCH models are extended with MLP, RBF, and RNN models and their modeling performances are tested.

*4.2.1. MS-ARMA-GARCH-Neural Networks Results.* In the study, model estimation is gathered through utilizing backpropagation algorithm and the parameters are updated with respect to a quadratic loss function, whereas the weights are iteratively calculated with *weight decay* method to achieve the lowest error. Alternative methods include Genetic Algorithms [132–134] and 2nd order derivative based optimization algorithms such as Conjugate Gradient Descent, Quasi-Newton, Quick Propagation, Delta-Bar-Delta, and Levenberg-Marquardt, which are fast and effective algorithms but may be subject to overfitting (see [135–137]). In the study, we followed a two-step methodology. Firstly, all models were trained over a given training sample vis-à-vis checking for generalization accuracy in light of RMSE criteria in test sample. The approach is repeated for estimating each model for 100 times with different number of sigmoid activation functions in the hidden layer. Hence, to obtain parsimony in models, best model is further selected with respect to the AIC information criterion. For estimating NN-GARCH models with early stopping combined with algorithm corporation, readers are referred to Bildirici and Ersin [61].

The estimated models are reported in Table 4. For comparative purposes, MSE and RMSE values for training sample are given. Among the MS-ARMA-GARCH-NN models and for the training sample, the lowest RMSE value is achieved as 0.114 by the MS-ARMA-GARCH-RBF model, followed by MS-ARMA-GARCH-MLP model with a RMSE value of 0.186 as the second best model among the models with GARCH specifications noted at the first part of Table 4. Furthermore, the above mentioned models are followed by MS-ARMA-GARCH-HYBRID MLP model that deserves the 3rd place with RMSE = 0.187, though the value is almost equal to the value obtained for the MS-ARMA-GARCH-MLP model having the 2nd place. Further, MS-ARMA-GARCH-RNN, and MS-ARMA-GARCH-ELMAN RNN models take the 4th and 5th place among the GARCH type competing models in terms of in-sample accuracy only.

The Markov-switching models with asymmetric power terms in the conditional volatility, namely, the, MS-ARMA-APGARCH-NN models are reported in the second part of Table 4. The model group consists of MS-ARMA-APGARCH-RNN, MS-ARMA-APGARCH-RBF, MS-ARMA-APGARCH-ELMAN RNN, and MS-ARMA-APGARCH-MLP models. It is observed among the group that the lowest RMSE value is attained for the MS-ARMA-APGARCH-MLP model with a value of 0.16307 followed by MS-ARMA-APGARCH-RNN model. Further, MS-ARMA-APGARCH-ELMAN RNN and MS-ARMA-APGARCH-RNN models possess the 2nd and 3rd places with RMSE values equal to 0.1663 and 0.1776, respectively.

Table 3: MS-ARMA-GARCH models.

| MS-ARMA-GARCH models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| (1) MS-ARMA-GARCH | Garch | Sigma | Arch | Constant | $p\_\{0 \mid 0\}$ | $P\_\{1 \mid 1\}$ | | | $\log L$ | RMSE |
| Regime 1. | 0.966483<br>$(0.01307)^{***}$ | 0.000333727<br>$(1.916e{-}006)^{***}$ | 0.03351<br>(0.005) | $6.23008e - 005$<br>$(1.360e{-}005)^{***}$ | 0.500244 | 0.5102 | | | 385.09 | 0.458911 |
| Regime 2. | 0.436124<br>$(0.01307)^{***}$ | 0.0004356<br>$(1.231e{-}005)^{***}$ | 0.56387<br>(0.0098) | $6.29344e - 005$<br>$(1.161e{-}005)^{***}$ | | | | | | |
| (2) MS-ARMA-APGARCH | Garch | sigma | Arch | Constant | Mean | $P\_\{0 \mid 0\}$ | $p\_\{1 \mid 1\}$ | Power | $\log L$ | RMSE |
| Regime 1. | 0.616759<br>$(0.01307)^{***}$ | 0.000679791<br>$(5.685e{-}006)^{***}$ | 0.383241<br>(0.0102) | $8.13394e - 005$<br>$(2.007e{-}005)^{***}$ | 0.00021042 | 0.500227 | 0.50300 | 0.80456 (0.00546) | 1756.5 | 0.42111 |
| Regime 2. | 0.791950<br>$(0.01307)^{***}$ | 0.0012381<br>$(3.451e{-}004)^{***}$ | 0.20805<br>(0.0201) | $8.20782e - 005$<br>$(1.835e{-}005)^{***}$ | $2.83E - 03$ | | | 0.60567 (0.0234) | | |
| (3) MS-ARMA-FIAPGARCH | ARCH<br>(Phi1) | GARCH<br>(Beta1) | d-Figarch | APARCH<br>(Gamma1) | APARCH<br>(Delta) | Cst | $p\_\{0 \mid 0\}$ | $p\_\{0 \mid 1\}$ | $\log L$ | RMSE |
| Regime 1. | 0.277721<br>(0.00) | 0.67848<br>(0.00) | 0.2761233<br>(0.0266) | 0.220157<br>(0.0106) | 0.123656 (0.001) | 0.00135<br>(0.001) | 0.50212 | 0.510021 | 1877.9 | 0.4222066 |
| Regime 2. | 0.309385<br>(0.002) | 0.680615<br>(0.0001) | 0.181542<br>(0.00005) | 0.21083<br>(0.0299) | 0.1448<br>(0.0234) | 0.00112<br>(0.00984) | | | | |

Table 4: Markov switching GARCH neural network models: training results.

| | MSE | RMSE |
|---|---|---|
| Model Group 1: MS-ARMA-GARCH-neural network models | | |
| MS-ARMA-GARCH-RNN | 0.035103533 | 0.187359368 (4th) |
| MS-ARMA-GARCH-RBF | 0.013009577 | 0.114059534 (1st) |
| MS-ARMA-GARCH-ELMAN RNN | 0.051024351 | 0.225885704 (5th) |
| MS-ARMA-GARCH-HYBRID MLP | 0.034996508 | 0.187073536 (3rd) |
| MS-ARMA-GARCH-MLP | 0.034665716 | 0.186187315 (2nd) |
| Model Group 2: MS-ARMA-APGARCH-neural network models | | |
| MS-ARMA-APGARCH-RNN | 0.031530707 | 0.177568881 (4th) |
| MS-ARMA-APGARCH-RBF | 0.056680761 | 0.238077218 (5th) |
| MS-ARMA-APGARCH-ELMAN RNN | 0.027644691 | 0.166266929 (3rd) |
| MS-ARMA-APGARCH-HYBRID MLP | 0.026504969 | 0.162803470 (1st) |
| MS-ARMA-APGARCH-MLP | 0.026591934 | 0.163070336 (2nd) |
| Model Group 3: MS-ARMA-FIAPGARCH-neural network models | | |
| MS-ARMA-FIAPGARCH-RNN | 0.029951509 | 0.173065042 (1st) |
| MS-ARMA-ARMA-FIAPGARCH-RBF | 0.045969206 | 0.214404305 (5th) |
| MS-ARMA-FIAPGARCH-ELMAN RNN | 0.033859273 | 0.184008896 (4th) |
| MS-ARMA-FIAPGARCH-HYBRID MLP | 0.031056323 | 0.176228044 (2nd) |
| MS-ARMA-FIAPGARCH-MLP | 0.031221803 | 0.176696926 (3rd) |

Lastly, the lowest RMSE value (=0.2381) is attained for the MS-APGARCH-RBF model among the group.

Models in the third model group, namely, MS-ARMA-FIAPGARCH-RNN, MS-ARMA-FIAPGARCH-RBF, MS-ARMA-FIAPGARCH-RNN, MS-ARMA-FIAPGARCH-HYBRID MLP, and MS-ARMA-FIAPGARCH-MLP models are modified versions of MS-ARMA-APGARCH-NN models with fractional integration. According to the results obtained in Table 4, the lowest RMSE value is obtained for the MS-ARMA-FIAPGARCH-RNN model (RMSE = 0.1731) followed by the MS-ARMA-APGARCH-HYBRID MLP model (RMSE = 0.1762). The RMSE is calculated as 0.1767 for the MS-ARMA-FIAPGARCH-MLP model, which obtained the 3rd place, though it is observed that the models that deserved the first three places have very close RMSE values. Further, MS-ARMA-FIAPGARCH-ELMAN RNN, and MS-ARMA-FIAPGARCH-RBF models possess the 4th and 5th places with RMSE values of 0.1840 and 0.2144, with respect to the results obtained for the training sample.

The results obtained for the test sample and the hold-out sample are given in Tables 5 and 6. If an overlook is to be provided, it is observed that holding the training sample results on one side, the real improvement occurs for forecasting. When we move from MS-ARMA-GARCH-NN models (denoted as Model Group 1) towards MS-ARMA-APGARCH (denoted as Model Group 2) and fractionally integrated models of MS-ARMA-FIAPGARCH-NN (Model Group 3) models, though certain amount of in-sample forecast gain is achieved, the MSE and RMSE values are the lowest for test sample and, most importantly, for the out-of-sample forecasts.

To evaluate the test sample performance of the estimated models, MSE and RMSE error criteria are reported in Table 5. Accordingly, among the MS-ARMA-GARCH-NN models, the lowest RMSE value for the test sample is 0.1238 and is calculated for the MS-ARMA-GARCH-MLP model that is followed by the MS-ARMA-GARCH-RNN model with a RMSE of 0.1242. The MS-ARMA-GARCH-HYBRID MLP model takes the 3rd place in the test sample, similar to the results obtained for the training sample with RMSE = 0.1243, though the value is almost equal to the models taking the first and the second places. Compared with the first three models we mentioned with RMSE values that are achieved as low as around 0.124, MS-ARMA-GARCH-ELMAN RNN, and MS-ARMA-GARCH-RBF models take the 4th and 5th places with RMSE values equal to 0.1538 and 0.1546. The results show that, for the test sample, MS-ARMA-GARCH-MLP, MS-ARMA-GARCH-RNN, and MS-ARMA-GARCH-HYBRID MLP models have very similar results and well generalization capabilities, though the last 2 competing models also have promising results. Furthermore, as we evaluate the APGARCH type models, the generalization capabilities increase significantly. Among the MS-ARMA-APGARCH-NN models given in the table, MS-ARMA-APGARCH-HYBRID MLP model has the lowest RMSE (=0.00011539) followed by the MS-ARMA-APGARCH-MLP model (RMSE = 0.00011548). The third best model is the *MS-ARMA-APGARCH-ELMAN RNN model with* RMSE = 0.00011789 followed by the 4th and 5th models; *MS-ARMA-APGARCH-RNN* (RMSE = 0.0001276) and *MS-ARMA-APGARCH-RBF* (RMSE = 0.0001631), respectively. As discussed, models in the third group are MS-ARMA-FIAPGARCH-NN models. According to the results obtained in Table 5, the lowest RMSE values are obtained for the first three models: MS-ARMA-FIAPGARCH-RNN (RMSE = 0.001164), MS-ARMA-APGARCH-HYBRID MLP (RMSE = 0.0001177), and MS-ARMA-FIAPGARCH-MLP (RMSE = 0.0001178) models. Similar to the results in the test sample, the models that

Table 5: Markov switching ARMA-GARCH neural network models: test sample results.

| | MSE | RMSE |
|---|---|---|
| Model Group 1: MS-ARMA-GARCH-neural network models | | |
| MS-ARMA-GARCH-RNN | 0.015436917 | 0.124245392 (2nd) |
| MS-ARMA-GARCH-RBF | 0.023914916 | 0.154644483 (5th) |
| MS-ARMA-GARCH-ELMAN RNN | 0.023667136 | 0.153841268 (4th) |
| MS-ARMA-GARCH-HYBRID MLP | 0.015461623 | 0.124344774 (3rd) |
| MS-ARMA-GARCH-MLP | 0.015333378 | 0.123828017 (1st) |
| Model Group 2: MS-ARMA-APGARCH-neural network models | | |
| MS-ARMA-APGARCH-RNN | 0.00000001628449 | 0.00012761067957 (4th) |
| MS-ARMA-APGARCH-RBF | 0.00000002662825 | 0.00016318165387 (5th) |
| MS-ARMA-APGARCH-ELMAN RNN | 0.00000001389858 | 0.00011789225071 (3rd) |
| MS-ARMA-APGARCH-HYBRID MLP | 0.00000001331575 | 0.00011539391162 (1st) |
| MS-ARMA-APGARCH-MLP | 0.00000001333791 | 0.00011548986313 (2nd) |
| Model Group 3: MS-ARMA-FIAPGARCH-neural network models | | |
| MS-ARMA-FIAPGARCH-RNN | 0.00000001354729 | 0.00011639284333 (1st) |
| MS-ARMA-FIAPGARCH-RBF | 0.00000002576468 | 0.00016051379977 (5th) |
| MS-ARMA-FIAPGARCH-ELMAN RNN | 0.00000001521219 | 0.00012333770222 (4th) |
| MS-ARMA-FIAPGARCH-HYBRID MLP | 0.00000001385396 | 0.00011770286713 (2nd) |
| MS-ARMA-FIAPGARCH-MLP | 0.00000001389543 | 0.00011787886146 (3rd) |

Table 6: Markov switching ARMA-GARCH neural network models: out of sample results.

| | MSE | RMSE |
|---|---|---|
| Model Group 1: MS-ARMA-GARCH-neural network models | | |
| MS-ARMA-GARCH | 0.210599000 | 0.458911000 (6th) |
| MS-ARMA-GARCH-RNN | 0.015436917 | 0.124245392 (2nd) |
| MS-ARMA-GARCH-RBF | 0.023914916 | 0.154644483 (5th) |
| MS-ARMA-GARCH-ELMAN RNN | 0.023667136 | 0.153841268 (4th) |
| MS-ARMA-GARCH-HYBRID MLP | 0.015461623 | 0.124344774 (3rd) |
| MS-ARMA-GARCH-MLP | 0.015333378 | 0.123828017 (1st) |
| Model Group 2: MS-ARMA-APGARCH-neural network models | | |
| MS-ARMA-APGARCH | 0.17740000000000 | 0.42111000000000 (6th) |
| MS-ARMA-APGARCH-RNN | 0.00000001628449 | 0.00012761067957 (4th) |
| MS-ARMA-APGARCH-RBF | 0.00000002662825 | 0.00016318165387 (5th) |
| MS-ARMA-APGARCH-ELMAN RNN | 0.00000001389858 | 0.00011789225071 (3rd) |
| MS-ARMA-APGARCH-HYBRID MLP | 0.00000001331575 | 0.00011539391162 (1st) |
| MS-ARMA-APGARCH-MLP | 0.00000001333791 | 0.00011548986313 (2nd) |
| Model Group 3: MS-ARMA-FIAPGARCH-neural network models | | |
| MS-ARMA-FIAPGARCH | 0.17814000000000 | 0.42220660000000 (6th) |
| MS-ARMA-FIAPGARCH-RNN | 0.00000001354729 | 0.00011639284333 (1st) |
| MS-ARMA-FIAPGARCH-RBF | 0.00000002576468 | 0.00016051379977 (5th) |
| MS-ARMA-FIAPGARCH-ELMAN RNN | 0.00000001521219 | 0.00012333770222 (4th) |
| MS-ARMA-FIAPGARCH-HYBRID MLP | 0.00000001385396 | 0.00011770286713 (2nd) |
| MS-ARMA-FIAPGARCH-MLP | 0.00000001389543 | 0.00011787886146 (3rd) |

deserved the first three places have very close RMSE values. Similarly, MS-ARMA-FIAPGARCH-ELMAN RNN (RMSE = 0.000123) and MS-ARMA-FIAPGARCH-RBF (RMSE = 0.00016) models took the 4th and 5th places.

In Table 6, the models are compared for their respective out-of-sample forecast performances. At the first part of Table 6 MS-ARMA-GARCH-MLP has the lowest RMSE (=0.1238) in the MS-GARCH-NN group followed by MS-ARMA-GARCH-RNN (RMSE = 0.1242) and MS-ARMA-GARCH-HYBRID MLP (RMSE = 0.1243) models. Similar to the results obtained for the test sample, the MS-ARMA-APGARCH-NN and MS-ARMA-FIAPGARCH-NN specifications show significant improvement in light of MSE and RMSE criteria. Among the MS-ARMA-APGARCH-NN models, the lowest RMSE is achieved by MS-ARMA-APGARCH-HYBRID MLP (RMSE =

0.00011539) followed by the MS-ARMA-APGARCH-MLP (RMSE = 0.00011548) model. MS-ARMA-APGARCH-ELMAN RNN and MS-ARMA-APGARCH-RNN models took the 4th and 5th places in out-of-sample forecasting; however, though significant improvement is reported, MS-ARMA-APGARCH-RBF model took the 5th place among other MS-ARMA-APGARCH-NN models in the group. If MS-ARMA-FIAPGARCH-NN models are evaluated, the MS-ARMA-FIAPGARCH-RNN and MS-ARMA-FIAPGARCH-HYBRID MLP models are observed to take the 1st and 2nd places with respective RMSE values of 0.0001163 and 0.0001177. Further, the MS-ARMA-FIAPGARCH-MLP model is the 3rd with RMSE = 0.0001178 and the 4th and the 5th are MS-ARMA-FIAPGARCH-ELMAN RNN (RMSE = 0.0001233) and MS-FIAPGARCH-RBF (RMSE = 0.0001605) models. It is noteworthy that though all RMSE values are significantly low, RBF based model has become the last model among the MS-ARMA-FIAPGARCH-NN group members. Further, Diebold-Mariano equal forecast accuracy tests will be applied to evaluate the models.

To evaluate the forecast accuracy of the estimated models, Diebold-Mariano test results are reported in Table 7. The forecasting sample corresponds to the last 587 observations of ISE100 daily return series. For interpretation purposes, [r] and [c] given in brackets denote that the selected model is the row or the column model with respect to the *P* value given in parentheses for the calculated Diebold-Mariano test statistic.

Firstly, among the models in Group 1, the MS-ARMA-GARCH-RNN model is compared with the MS-ARMA-GARCH-RBF model in the first column. The calculated DM test statistic is −4.020 and significant at the 1% significance level, thus, the null hypothesis of equal forecast accuracy is rejected in favor of MS-ARMA-GARCH-RNN model. The MS-ARMA-GARCH-RBF model is compared with the MS-ARMA-GARCH-RNN model. The DM statistic is −3.176 and shows that at 5% significance level MS-ARMA-GARCH-RNN model is selected over MS-ARMA-GARCH-ELMAN RNN model. If MS-ARMA-GARCH-HYBRID MLP and MS-ARMA-GARCH-RNN models are compared, we fail to reject the null hypothesis of equal forecast accuracy. If MS-ARMA-GARCH-MLP and MS-ARMA-GARCH-RNN models are compared, DM test statistic is calculated as 0.574 with *P* value = 0.56, therefore, the null hypothesis of equal forecast accuracy cannot be rejected. The results show that MS-ARMA-GARCH-RNN model is preferred over MS-ARMA-GARCH-RBF and MS-ARMA-GARCH-ELMAN RNN models. If the MS-ARMA-GARCH-RBF model at the second row is analyzed, the null of equal forecast accuracy cannot be rejected between MS-ARMA-GARCH-RBF and MS-ARMA-GARCH-ELMAN RNN models, whereas MS-ARMA-GARCH-HYBRID MLP (DM = 4.011) and MS-ARMA-GARCH-MLP (DM = 4.142) models are selected over MS-ARMA-GARCH-RBF model. The equal forecast accuracy of MS-ARMA-GARCH-ELMAN RNN and MS-ARMA-GARCH-HYBRID MLP models is rejected and the alternative hypothesis that MS-ARMA-GARCH-HYBRID MLP model has better forecast accuracy is accepted at 1% significance level (DM = 3.309). The DM statistic calculated for MS-ARMA-GARCH-ELMAN RNN and MS-ARMA-GARCH-MLP models is 3.396 suggesting that MS-GARCH-MLP model is selected.

The asymmetric power GARCH architecture based models in Group 2 are evaluated at the second part of Table 7. MS-ARMA-APGARCH-RNN is compared with the other models at the first row. DM test statistic is calculated as −2.932 suggesting that the null of equal forecast accuracy between MS-ARMA-APGARCH-RNN and MS-ARMA-APGARCH-RBF is rejected in favor of MS-ARMA-APGARCH-RNN model. Further, the null hypotheses for the test of equal forecast accuracy between MS-ARMA-APGARCH-RNN and the other models are rejected and better forecast accuracy of MS-ARMA-APGARCH-ELMAN RNN, MS-ARMA-APGARCH-HYBRID MLP, and MS-ARMA-APGARCH-MLP are accepted at 1% significance level. Similar to the analysis conducted before, RBF based MS-ARMA-APGARCH-RBF model fail to have better forecast accuracy than the other models at the second row. MS-ARMA-APGARCH-HYBRID MLP model proved to have better forecast accuracy than all other models, whereas MS-ARMA-APGARCH-MLP model show to provide better forecast accuracy than all other models except for the MS-ARMA-APGARCH-HYBRID MLP model at 1% significance level.

MS-ARMA-FIAPGARCH-NN results are given in the third section of Table 7. The MS-ARMA-FIAPGARCH-RNN model is compared with the MS-ARMA-FIAPGARCH-RBF model in the first column. The calculated DM test statistic is −2.519 and suggests that the null hypothesis of equal forecast accuracy is rejected in favor of MS-ARMA-FIAPGARCH-RNN model. The DM statistic is −1.717 and shows that MS-ARMA-FIAPGARCH-RNN model is selected over MS-ARMA-FIAPGARCH-ELMAN RNN model only at 10% significance level. If we compare the MS-ARMA-FIAPGARCH-HYBRID MLP and MS-ARMA-GARCH-RNN models, we fail to reject the null hypothesis. On the other hand, MS-FIAPGARCH-HYBRID MLP is preferred over MS-ARMA-FIAPGARCH-ELMAN RNN at 5% significance level (DM = 2.154) and over MS-ARMA-FIAPGARCH-RBF model at 1% significance level (DM = 2.608). Accordingly, MS-ARMA-FIAPGARCH-RBF model fail to beat other models though RMSE values show significant improvement, the MS-ARMA-FIAPGARCH-RNN model is selected over the other models, whereas the MS-ARMA-FIAPGARCH-HYBRID MLP model is the second best model for the analyzed time series and analyzed sample.

## 5. Conclusions

In the study, a family of regime switching neural network augmented volatility models is analyzed to achieve an evaluation of forecast accuracy and an application to daily returns in an emerging market stock index presented. In this respect, GARCH neural network models, which allow the generalization of MS type RS-GARCH models to MS-ARMA-GARCH-NN models are incorporated with neural networks models based on Multilayer Perceptron (MLP), Radial Basis Function (RBF), Elman Recurrent NN

Table 7: Diebold Mariano equal forecast accuracy test results, out of sample.

| | Model Group 1: MS-ARMA-GARCH-neural network models | | | | |
|---|---|---|---|---|---|
| | MS-ARMA-GARCH-RNN | MS-ARMA-GARCH-RBF | MS-ARMA-GARCH-ELMAN RNN | MS-ARMA-GARCH-HYBRID MLP | MS-ARMA-GARCH-MLP |
| MS-ARMA-GARCH-RNN | — | $-4.020^{***}$ (0.000) [r] | $-3.176^{**}$ (0.002) [r] | −0.645 (0.518) [r] | 0.574 (0.566) [c] |
| MS-ARMA-GARCH-RBF | | — | 0.585 (0.558) [c] | $4.011^{***}$ (0.000) [c] | $4.142^{***}$ (0.000) [c] |
| MS-ARMA-GARCH-ELMAN RNN | | | — | $3.309^{***}$ (0.001) [c] | $3.396^{***}$ (0.001) [c] |
| MS-ARMA-GARCH-HYBRID MLP | | | | — | $3.355^{***}$ (0.01) [c] |
| MS-ARMA-GARCH-MLP | | | | | — |
| | Model Group 2: MS-ARMA-APGARCH-neural network models | | | | |
| | MS-ARMA-APGARCH-RNN | MS-ARMA-APGARCH-RBF | MS-ARMA-APGARCH-ELMAN RNN | MS-ARMA-APGARCH-HYBRID MLP | MS-ARMA-APGARCH-MLP |
| MS-ARMA-APGARCH-RNN | — | $-2.932^{***}$ (0.003) [r] | $3.767^{***}$ (0.000) [c] | $4.888^{***}$ (0.000) [c] | $4.805^{***}$ (0.000) [c] |
| MS-ARMA-APGARCH-RBF | | — | $3.188^{***}$ (0.001) [c] | $3.251^{***}$ (0.001) [c] | $3.255^{***}$ (0.001) [c] |
| MS-ARMA-APGARCH-ELMAN RNN | | | — | $2.797^{***}$ (0.005) [c] | $2.736^{***}$ (0.006) [c] |
| MS-ARMA-APGARCH-HYBRID MLP | | | | — | $-1.835^{**}$ (0.066) [r] |
| MS-ARMA-APGARCH-MLP | | | | | — |
| | Model Group 3: MS-ARMA-FIAPGARCH-neural network models | | | | |
| | MS-ARMA-FIAPGARCH-RNN | MS-ARMA-FIAPGARCH-RBF | MS-ARMA-FIAPGARCH-ELMAN RNN | MS-ARMA-FIAPGARCH-HYBRID MLP | MS-ARMA-FIAPGARCH-MLP |
| MS-ARMA-FIAPGARCH-RNN | — | $-2.519^{**}$ (0.011) [r] | $-1.717^{*}$ (0.086) [r] | −0.588 (0.556) [r] | −0.836 (0.403) [r] |
| MS-ARMA-FIAPGARCH-RBF | | — | $2.214^{**}$ (0.027) [c] | $2.608^{***}$ (0.009) [c] | $2.526^{**}$ (0.012) [c] |
| MS-ARMA-FIAPGARCH-ELMAN RNN | | | — | $2.154^{**}$ (0.031) [c] | $-2.214^{**}$ (0.026) [r] |
| MS-ARMA-FIAPGARCH-HYBRID MLP | | | | — | −0.716 (0.473) [r] |
| MS-ARMA-FIAPGARCH-MLP | | | | | — |

Notes: Statistical significances of the relevant tests are denoted with $^{***}$show statistical significance at 1% significance level; while $^{**,and*}$show significance at 5% and 10%, respectively. D-M test allows for reporting the selected model. Accordingly, [r] shows that the model reported in the "row" is selected over the model in the column. Similarly, [c] stands for the column model being accepted over the row model.

(Elman RNN), Time Lag (delay) Recurrent NN (RNN), and Hybrid MLP models. Gray [92] RS-GARCH model is taken as the baseline modeling structure in the study. Gray [92] utilizes the Hamilton [67] Markov-switching approach, however the regime-switching is allowed in the conditional volatility processes of a time series. On the other hand, the study aimed at Hamilton [67] type regime-switching between GARCH-Neural Network approaches developed by Donaldson and Kamstra [118] and further extanded to a class of GARCH-NN models by Bildirici and Ersin [61]. The model within the family analyzed are MS-ARMA-GARCH-MLP, MS-ARMA-GARCH-RBF, MS-ARMA-GARCH-ElmanRNN, MS-ARMA-GARCH-RNN, and MS-ARMA-GARCH-Hybrid MLP which are further extended to account for asymmetric power terms based on the APGARCH architecture, MS-ARMA-APGARCH-MLP, MS-ARMA-APGARCH-RBF, MS-ARMA-APGARCH-ElmanRNN, MS-ARMA-APGARCH-RNN, and MS-ARMA-APGARCH-Hybrid MLP.

Further, by accounting for fractional integration (FI) in GARCH specification, the models are generalized as MS-ARMA-FIAPGARCH-MLP, MS-ARMA-FIAPGARCH-RBF, MS-ARMA-FIAPGARCH-ElmanRNN, MS-ARMA-FIAPGARCH-RNN, and MS-ARMA-FIAPGARCH-Hybrid MLP models that account for fractionally integrated models with asymmetric power transformations and generalized to neural network models with possibly improved forecast capabilities.

Models are evaluated with MSE and RMSE error criteria. To evaluate equal forecast accuracy, modified Diebold-Mariano tests are applied. It is observed that, holding the training sample results on one side, the real improvement occurs for the test sample and most importantly, for the out-of-sample. By evaluating MS-GARCH-NN models, though in-sample performance is noticeable, moving towards MS-ARMA-APGARCH-NN models and fractionally integrated models of MS-ARMA-FIAPGARCH-NN, significant improvement is noticed in light of the MSE and RMSE criteria and in terms of Diebold-Mariano equal forecast accuracy tests.

Among the models analyzed, asymmetric power modeling with fractional integration generalized to Time Lag Recurrent Neural Network architecture and Hybrid Multilayer Perceptron are shown to provide significant forecast and modeling capabilities. Thus, the models with regime switching further augmented with neural network modeling techniques promise significant achievements in return and volatility modeling and forecasting.

## Conflict of Interests

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

## References

[1] C. Brooks, "Predicting stock index volatility: can Market Volume help?" *Journal of Forecasting*, vol. 17, pp. 59–98, 1998.

[2] J. Y. Campbell, S. J. Grossman, and J. Wang, "Trading volume and serial correlation in stock returns," *The Quarterly Journal of Economics*, vol. 108, pp. 905–936, 1993.

[3] C. Hiemstra and J. D. Jones, "Testing for linear and nonlinear Granger causality in the stock price-volume relation," *The Journal of Finance*, vol. 49, pp. 1639–1664, 1994.

[4] X. Wang, P. H. Phua, and W. Lin, "Stock market prediction using neural networks: does trading volume help in short-term prediction?" in *Proceedings of IEEE International Joint Conference on Neural Networks,*, vol. 4, pp. 2438–2442, 2003.

[5] R. F. Engle, "Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, pp. 987–1007, 1982.

[6] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[7] R. F. Engle and T. Bollerslev, "Modeling the persistence of conditional variances," *Econometric Reviews*, vol. 5, no. 1, pp. 1–50, 1986.

[8] R. F. Engle, "Discussion on schwert," *The Review of Financial Studies*, vol. 3, pp. 103–106, 1990.

[9] D. B. Nelson, "Conditional heteroscedasticity in assets returns: a new approach," *Econometrica*, vol. 55, pp. 703–708, 1991.

[10] L. R. Glosten, R. Jagannathan, and D. Runkle, "On the relation between the expected value and the volatility of the nominal excess return on stocks," *Journal of Finance*, vol. 48, pp. 1779–1801, 1993.

[11] J.-M. Zakoian, "Threshold heteroskedastic models," *Journal of Economic Dynamics and Control*, vol. 18, no. 5, pp. 931–955, 1994.

[12] J. M. Zakoian, "Threshold heteroscedastic models," Discussion Paper, INSE, 1991.

[13] R. F. Engle and V. K. Ng, "Measuring and testing the impact of news on volatility," *Journal of Finance*, vol. 48, pp. 1749–1778, 1993.

[14] S. L. Heston and S. Nandi, "A closed-form GARCH option valuation model," *Review of Financial Studies*, vol. 13, no. 3, pp. 585–625, 2000.

[15] J.-C. Duan, "Augmented GARCH(p,q) process and its diffusion limit," *Journal of Econometrics*, vol. 79, no. 1, pp. 97–127, 1997.

[16] E. Sentana, "Quadratic ARCH Models: a potential reinterpretation of ARCH Models as second-order Taylor approximations," Working Paper, London School of Economics, 1991.

[17] R. T. Baillie, T. Bollerslev, and H. O. Mikkelsen, "Fractionally integrated generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 74, no. 1, pp. 3–30, 1996.

[18] T. Bollerslev, "Glossary to ARCH (GARCH)," 2009, http://public.econ.duke.edu/~boller/Papers/glossary_arch.pdf.

[19] Z. Ding, C. W. J. Granger, and R. F. Engle, "A long memory property of stock market returns and a new model," *Journal of Empirical Finance*, vol. 1, no. 1, pp. 83–106, 1993.

[20] Y. K. Tse, "The conditional heteroscedasticity of the yen-dollar exchange rate," *Journal of Applied Econometrics*, vol. 13, no. 1, pp. 49–55, 1998.

[21] J. Davidson, *Stochastic Limit Theory*, Oxford University Press, New York, NY, USA, 1994.

[22] G. Perez-Quiros and A. Timmermann, "Business cycle asymmetries in stock returns: evidence from higher order moments and conditional densities," *Journal of Econometrics*, vol. 103, no. 1-2, pp. 259–306, 2001.

[23] F. Diebold, "Comment on modelling the persistence of conditional variances," *Econometric Reviews*, vol. 5, pp. 51–56, 1986.

[24] C. Lamoureux and W. Lastrapes, "Persistence in variance, structural change, and the GARCH model," *Journal of Business and Economic Statistics*, vol. 8, pp. 225–234, 1990.

[25] Z. Qiao and W. Wong, "Revisiting volume vs. GARCH effects using univariate and bivariate GARCH models: evidence from U.S. Stock markets," in *Handbook of Quantitative Finance and Risk Management*, C. Lee, A. Lee, and J. Lee, Eds., part 5, pp. 1173–1181, Springer, 2010.

[26] T. Mikosch and C. Stărică, "Nonstationarities in financial time series, the long-range dependence, and the IGARCH effects," *Review of Economics and Statistics*, vol. 86, no. 1, pp. 378–390, 2004.

[27] L. Bauwens, A. Preminger, and J. Rombouts, Theory and Inference for a Markov Switching GARCH Model, Center for 2009-11.

[28] L. Bauwens, A. Preminger, and J. Rombouts, "Regime Switching GARCH Models," CORE Discussion Paper, Universite Catholique de Louvain, Louvain La Neuve 2006/11, 2006.

[29] W. Krämer, "Long Memory with Markov-Switching GARCH," Cesifo WP 2225, 2008.

[30] C. Francq and J.-M. Zakoïan, "The L2-structures of standard and switching-regime GARCH models," *Stochastic Processes and their Applications*, vol. 115, no. 9, pp. 1557–1582, 2005.

[31] C. Alexander and E. Lazaar, "Markov switching GARCH diffusion," ICMA Centre Discussion Papers in Finance 2008-01, 2008.

[32] W. Krämer and B. Tameze, "Structural change and estimated persistence in the GARCH(1,1)-model," SFB 475 Discussion Paper 33/06, Universität Dortmund, 2006.

[33] C. M. Turner, R. Startz, and C. R. Nelson, "A Markov model of heteroskedasticity, risk, and learning in the stock market," *Journal of Financial Economics*, vol. 25, no. 1, pp. 3–22, 1989.

[34] P. Taylor, "Hidden Markov models for grapheme to phoneme conversion," in *Proceedings of the 9th European Conference on Speech Communication and Technology*, pp. 1973–1976, September 2005.

[35] C. Alexander and A. Dimitriu, "Indexing, cointegration and equity market regimes," *International Journal of Finance and Economics*, vol. 10, no. 3, pp. 213–231, 2005.

[36] Y.-W. Cheung and U. G. Erlandsson, "Exchange rates and markov switching dynamics," *Journal of Business and Economic Statistics*, vol. 23, no. 3, pp. 314–320, 2005.

[37] N. Francis and M. T. Owyang, "Monetary policy in a markov-switching vector error-correction model: implications for the cost of disinflation and the price puzzle," *Journal of Business and Economic Statistics*, vol. 23, no. 3, pp. 305–313, 2005.

[38] R. H. Clarida, L. Sarno, M. P. Taylor, and G. Valente, "The role of asymmetries and regime shifts in the term structure of interest rates," *Journal of Business*, vol. 79, no. 3, pp. 1193–1224, 2006.

[39] C. Alexander and A. Kaeck, "Regimes in CDS spreads: a markov switching model of iTraxx Europe indices," ICMA Centre Discussion Papers in Finance icma-dp2006-08, Henley Business School, Reading University, 2006.

[40] A. Kanas, "Non-linear forecasts of stock returns," *Journal of Forecasting*, vol. 22, no. 4, pp. 299–315, 2003.

[41] T. L. Lai and S. P.-S. Wong, "Stochastic neural networks with applications to nonlinear time series," *Journal of the American Statistical Association*, vol. 96, no. 455, pp. 968–981, 2001.

[42] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart, "Predicting sunspots and exchange rates with connectionist networks," in *Proceedings of the NATO Workshop on Nonlinear Modeling and Forecasting*, Addison Wesley, Santa Fe, NM, USA, 1991.

[43] A. S. Weigend and N. A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1993.

[44] H. White, *Artificial Neural Networks: Approximation and Learning Theory*, Blackwell, Oxford, UK, 1992.

[45] J. M. Hutchinson, A. W. Lo, and T. Poggio, "A nonparametric approach to pricing and hedging derivative securities via learning networks," *The Journal of Finance*, vol. 49, no. 3, pp. 851–889, 1994.

[46] A.-P. N. Refenes, A. N. Burgess, and Y. Bentz, "Neural networks in financial engineering: a study in methodology," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1222–1267, 1997.

[47] H. Wang, V. Jacob, and E. Rolland, "Design of efficient hybrid neural networks for flexible flow shop scheduling," *Expert Systems*, vol. 20, no. 4, pp. 208–231, 2003.

[48] H. Wang, "Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions," *Expert Systems*, vol. 22, no. 2, pp. 78–85, 2005.

[49] A. Abhyankar, L. S. Copeland, and W. Wong, "Uncovering nonlinear structure in real-time stock-market indexes: the S&P 500, the DAX, the Nikkei 225, and the FTSE-100," *Journal of Business and Economic Statistics*, vol. 15, no. 1, pp. 1–14, 1997.

[50] F. Castiglione, "Forecasting price increments using an artificial neural network," *Advances in Complex Systems*, vol. 4, no. 1, pp. 45–56, 2001.

[51] B. Freisleben, "Stock market prediction with back propagation networks," in *Proceedings of the 5th International Conference on Industrial and Engineering Application of Artificial Intelligence and Expert System*, pp. 451–460, 1992.

[52] S. H. Kim and S. H. Chun, "Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index," *International Journal of Forecasting*, vol. 14, no. 3, pp. 323–337, 1998.

[53] Y. Liu and X. Yao, "Evolving neural networks for Hang Seng stock index forecast," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 256–260, May 2001.

[54] P. K. H. Phua, X. Zhu, and C. H. Koh, "Forecasting stock index increments using neural networks with trust region methods," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 260–265, July 2003.

[55] A. N. Refenes, A. Zapranis, and G. Francies, "Stock performance modeling using neural networks: a comparative study with regression models," *Neural Networks*, vol. 5, pp. 961–970, 1994.

[56] M. Resta, "Towards an artificial technical analysis of financial markets," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '2000)*, pp. 117–122, July 2000.

[57] R. Sitte and J. Sitte, "Analysis of the predictive ability of time delay neural networks applied to the S&P 500 time series," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 30, no. 4, pp. 568–572, 2000.

[58] P. Tiňo, C. Schittenkopf, and G. Dorffner, "Financial volatility trading using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865–874, 2001.

[59] J. T. Yao and H. L. Poh, "Equity forecasting: a case study on the KLSE index," in *Neural Networks in Financial Engineering, Proceedings of the 3rd International Conference on Neural Networks in the Capital Markets*, A. -P. N. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend, Eds., pp. 341–353, World Scientific, 1996.

[60] J. T. Yao and C. L. Tan, "Time dependent directional profit model for financial time series forecasting," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '00)*, pp. 291–296, July 2000.

[61] M. Bildirici and Ö. Ö. Ersin, "Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7355–7362, 2009.

[62] P. Ou and H. Wang, "Financial volatility forecasting by least square support vector machine based on GARCH, EGARCH and GJR models: evidence from ASEAN stock markets," *International Journal of Economics and Finance*, vol. 2, no. 1, pp. 51–64, 2010.

[63] A. Azadeh, M. Saberi, and M. Anvari, "An integrated artificial neural network algorithm for performance assessment and optimization of decision making units," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5688–5697, 2010.

[64] A. Bahrammirzaee, “A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems,” *Neural Computing and Applications*, vol. 19, no. 8, pp. 1165–1195, 2010.

[65] A. Kanas and A. Yannopoulos, “Comparing linear and nonlinear forecasts for stock returns,” *International Review of Economics and Finance*, vol. 10, no. 4, pp. 383–398, 2001.

[66] G. Schwert, “Why does stock market volatility change over time?” *Journal of Finance*, vol. 44, pp. 1115–1153, 1989.

[67] J. D. Hamilton, “A new approach to the economic analysis of nonstationary time series and the business cycle,” *Econometrica*, vol. 57, pp. 357–384, 1989.

[68] J. D. Hamilton, *Time Series Analysis*, Princeton University Press, Princeton, NJ, USA, 1994.

[69] J. D. Hamilton, “Specification testing in markov-switching time-series models,” *Journal of Econometrics*, vol. 70, pp. 127–157, 1996.

[70] J. D. Hamilton and L. Gang, “Stock market volatility and the business cycle,” *Journal of Applied Econometrics*, vol. 11, pp. 573–593, 1996.

[71] J. D. Hamilton, “Regime-Switching Models,” 2005, http://dss.ucsd.edu/~jhamilto/palgrav1.pdf.

[72] J. D. Hamilton, “Analysis of time series subject to changes in regime,” *Journal of Econometrics*, vol. 45, no. 1-2, pp. 39–70, 1990.

[73] H. M. Krolzig, *Markov Switching Vector Autoregressions. Modelling, Statistical Inference and Application to Business Cycle Analysis*, Springer, Berlin, Germany, 1997.

[74] H. M. Krolzig, *Econometric Modelling of Markov-Switching Vector Autoregressions Using MSVAR for Ox*, Oxford University, 1998.

[75] H. M. Krolzig, “Predicting markov-switching vector autoregressive processes,” in *Working Paper W31*, Oxford University, 2000.

[76] H. M. Krolzig, *Estimation, Structural Analysis and Forecasting of Regime-Switching Model with MSVAR for Ox*, Oxford University, 2001.

[77] H.-M. Krolzig and J. Toro, “Classical and modern business cycle measurement: the European case,” *Spanish Economic Review*, vol. 7, no. 1, pp. 1–21, 2005.

[78] C. Francq, M. Roussignol, and J.-M. Zakoïan, “Conditional heteroskedasticity driven by hidden Markov chains,” *Journal of Time Series Analysis*, vol. 22, no. 2, pp. 197–220, 2001.

[79] C. S. -Wong and W. K. Li, “On a mixture autoregressive conditional heteroscedastic model,” *Journal of American Statistical Association*, vol. 96, no. 455, pp. 982–995, 2001.

[80] C. Alexander and E. Lazaar, “The equity index skew, market crashes and asymmetric normal mixture GARCH,” ISMA Centre Discussion papers in Finance, 2004, http://www.icmacentre.ac.uk/.

[81] M. Haas, S. Mittnik, and M. Paolella, “A new approach to Markov-switching GARCH models,” *Journal of Financial Econometrics*, vol. 2, pp. 493–530, 2004.

[82] M. Haas, S. Mittnik, and M. Paolella, “Mixed normal conditional heteroskedasticity,” *Journal of Financial Econometrics*, vol. 2, pp. 211–250.

[83] “The skew-normal Markov-switching GARCH process,” Haas, 2004, http://www.socialpolitik.ovgu.de/sozialpolitik_media/papers/Haas_Markus_uid219_pid176.pdf.

[84] W. Scherrer and E. Ribarits, “On the parametrization of multivariate garch models,” *Econometric Theory*, vol. 23, no. 3, pp. 464–484, 2007.

[85] F. Klaassen, “Improving GARCH volatility forecasts with regime-switching GARCH,” *Empirical Economics*, vol. 27, no. 2, pp. 363–394, 2002.

[86] J. Cai, “A markov model of switching-regime ARCH,” *Journal of Business and Economic Statistics*, vol. 12, pp. 309–316, 1994.

[87] S. Kaufman and S. Frühwirth-Schnatter, “Bayesian analysis of switching ARCH models,” *Journal of Time Series Analysis*, vol. 23, pp. 425–458, 2002.

[88] S. Kaufmann and M. Scheicher, “A switching ARCH model for the German DAX index,” *Studies in Nonlinear Dynamics and Econometrics*, vol. 10, no. 4, 2006.

[89] D. Das and B. H. Yoo, “A Bayesian MCMC Algorithm for Markov Switching GARCH models,” Econometric Society 2004 North American Summer Meetings 179, Econometric Society, 2004.

[90] L. Bauwens, A. Preminger, and J. V. K. Rombouts, “Theory and inference for a Markov switching GARCH model,” *Econometrics Journal*, vol. 13, no. 2, pp. 218–244, 2010.

[91] J. D. Hamilton and R. Susmel, “Autoregressive conditional heteroskedasticity and changes in regime,” *Journal of Econometrics*, vol. 64, no. 1-2, pp. 307–333, 1994.

[92] S. F. Gray, “Modeling the conditional distribution of interest rates as a regime-switching process,” *Journal of Financial Economics*, vol. 42, no. 1, pp. 27–62, 1996.

[93] M. Yang, “Some properties of vector autoregressive processes with markov-switching coefficients,” *Econometric Theory*, vol. 16, no. 1, pp. 23–43, 2000.

[94] J.-F. Yao and J.-G. Attali, “On stability of nonlinear AR processes with Markov switching,” *Advances in Applied Probability*, vol. 32, no. 2, pp. 394–407, 2000.

[95] J. Yao, “On square-integrability of an AR process with Markov switching,” *Statistics and Probability Letters*, vol. 52, no. 3, pp. 265–270, 2001.

[96] C. Francq and J.-M. Zakoïan, “Comments on the paper by Minxian Yang: ‘Some properties of vector autoregressive processes with Markov-switching coefficients,’” *Econometric Theory*, vol. 18, no. 3, pp. 815–818, 2002.

[97] M. Haas, “Skew-normal mixture and Markov-switching GARCH processes,” *Studies in Nonlinear Dynamics and Econometrics*, vol. 14, no. 4, pp. 1–56, 2010.

[98] C.-H. Chang, J.-J. Lin, N. Pal, and M.-C. Chiang, “A note on improved approximation of the binomial distribution by the skew-normal distribution,” *American Statistician*, vol. 62, no. 2, pp. 167–170, 2008.

[99] J.-C. Liu, “Stationarity for a Markov-switching Box-Cox transformed threshold GARCH process,” *Statistics and Probability Letters*, vol. 77, no. 13, pp. 1428–1438, 2007.

[100] A. Abramson and I. Cohen, “On the stationarity of Markov-switching garch processes,” *Econometric Theory*, vol. 23, no. 3, pp. 485–500, 2007.

[101] M. J. Dueker, “Markov switching in GARCH processes and mean-reverting stock-market volatility,” *Journal of Business and Economic Statistics*, vol. 15, no. 1, pp. 26–34, 1997.

[102] C.-J. Kim, “Dynamic linear models with Markov-switching,” *Journal of Econometrics*, vol. 60, no. 1-2, pp. 1–22, 1994.

[103] M. Yang, “Some properties of vector autoregressive processes with markov-switching coefficients,” *Econometric Theory*, vol. 16, no. 1, pp. 23–43, 2000.

[104] L. Bauwens and J. V. K. Rombouts, “Bayesian inference for the mixed conditional heteroskedasticity model,” *Econometrics Journal*, vol. 10, no. 2, pp. 408–425, 2007.

[105] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[106] M. Frömmel, "Volatility regimes in central and Eastern European countries' exchange rates," Faculteit Economie en Bedrijfskunde Working Paper 2007/487.

[107] S. I. Blazsek and A. Downarowicz, "Regime switching models of hedge fund returns," Working Paper, 2008.

[108] J. S. Henneke, S. T. Rachev, F. J. Fabozzi, and M. Nikolove, "MCMC-based estimation of Markov Switching ARMA-GARCH models," *Applied Economics*, vol. 43, no. 3, pp. 259–271, 2011.

[109] M. Haas, "The autocorrelation structure of the Markov-switching asymmetric power GARCH process," *Statistics and Probability Letters*, vol. 78, no. 12, pp. 1480–1489, 2008.

[110] M. Haas, S. Mittnik, and M. S. Paolella, "Asymmetric multivariate normal mixture GARCH. Forthcoming," in *Computational Statistics and Data Analysis*, 2008.

[111] T. G. Andersen and T. Bollerslev, "Heterogeneous information arrivals and return volatility dynamics: uncovering the long-run in high frequency returns," *Journal of Finance*, vol. 52, no. 3, pp. 975–1005, 1997.

[112] Y. K. Tse, "The conditional heteroscedasticity of the yen-dollar exchange rate," *Journal of Applied Econometrics*, vol. 13, no. 1, pp. 49–55, 1998.

[113] L. Spezia and R. Paroli, "Bayesian inference and forecasting in dynamic neural networks with fully Markov switching ARCH noises," *Communications in Statistics*, vol. 37, no. 13, pp. 2079–2094, 2008.

[114] S. Dutta and S . Shektar, "Bond ratings: a non-conservative application of neural network," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 2, pp. 43–450, 1998.

[115] K. Y. Tom and M. Y. Kiang, "Managerial applications of neural networks: the case of bank failure predictions," *Management Science*, vol. 38, no. 7, pp. 926–947, 1992.

[116] A. Do and G. Grudinsky, "A neural network approach to residential property appraisal," in *The Real Estate Appraiser (December 1992)*, pp. 38–45, 1992.

[117] P. A. Shively, "The nonlinear dynamics of stock prices," *Quarterly Review of Economics and Finance*, vol. 43, no. 3, pp. 505–517, 2003.

[118] R. G. Donaldson and M. Kamstra, "Forecast combining with neural networks," *Journal of Forecasting*, vol. 15, no. 1, pp. 49–61, 1996.

[119] M. Olteanu, J. Rynkiewicz, and B. Maillet, "Nonlinear analysis of shocks when financial markets are subject to changes in regime," in *ESANN*, pp. 28–30, April 2004.

[120] D. Liu and L. Zhang, "China stock market regimes prediction with artificial neural network and markov regime switching," in *World Congress on Engineering (WCE '10)*, vol. 1, pp. 378–383, London, UK, July 2010.

[121] A. A. P. Santos, L. D. S. Coelho, and C. E. Klein, "Forecasting electricity prices using a RBF neural network with GARCH errors," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '10)*, pp. 1–8, July 2010.

[122] M. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," in *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, pp. 531–545, 1986.

[123] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[124] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[125] J. M. Binner, C. T. Elger, B. Nilsson, and J. A. Tepper, "Predictable non-linearities in U.S. Inflation," *Economics Letters*, vol. 93, no. 3, pp. 323–328, 2006.

[126] P. Tiňo, C. Schittenkopf, and G. Dorffner, "Financial volatility trading using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865–874, 2001.

[127] M. McKenzie and H. Mitchell, "Generalized asymmetric power ARCH modelling of exchange rate volatility," *Applied Financial Economics*, vol. 12, no. 8, pp. 555–564, 2002.

[128] R. B. Davies, "Hypothesis testing when a nuisance parameter is present only under the alternative," *Biometrika*, vol. 74, no. 1, pp. 33–43, 1987.

[129] M. Ural, "Generalized Asymmetric Power ARCH Modelling of National Stock Market Returns, SÜ İİBF Sosyal ve Ekonomik Araştırmalar Dergisi," pp. 575–590, 2009, http://www.iibf.selcuk.edu.tr/iibf_dergi/dosyalar/851348078522.pdf.

[130] T. Ané and L. Ureche-Rangau, "Stock market dynamics in a regime-switching asymmetric power GARCH model," *International Review of Financial Analysis*, vol. 15, no. 2, pp. 109–129, 2006.

[131] E. Teletar and Ş. Binay, "IMKB Endeksinin Üslü Otoregresif Koşullu Değişken Varyans (PARCH) ile Modellenmesi," 2001, http://idari.cu.edu.tr/sempozyum/bil6.htm.

[132] D. E. Goldberg, "Genetic algorithms and Walsh functions: part II, deception and its analysis," *Complex System*, vol. 3, pp. 153–171, 1989.

[133] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[134] C. D. E. Goldberg, "Sizing populations for serial and parallel genetic algorithms," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 70–79, 1989.

[135] D. Patterson, *Artifical Neural Networks*, Prentice Hall, Singopure, 1996.

[136] S. Haykin, *Neural Networks. A Comprehensive Foundation*, Macmillan, New York, NY, USA, 1994.

[137] L. Fausett, *Fundamentals of Neural Networks*, Prentice Hall, Englewood Cliffs, NJ, USA, 1994.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 798129, 23 pages
http://dx.doi.org/10.1155/2014/798129

*Research Article*

# Improved Particle Swarm Optimization with a Collective Local Unimodal Search for Continuous Optimization Problems

**Martins Akugbe Arasomwan and Aderemi Oluyinka Adewumi**

*School of Mathematics, Statistics, and Computer Science, University of Kwazulu-Natal South Africa,
Private Bag X54001, Durban 4000, South Africa*

Correspondence should be addressed to Aderemi Oluyinka Adewumi; laremtj@gmail.com

Received 31 October 2013; Accepted 29 December 2013; Published 25 February 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

A new local search technique is proposed and used to improve the performance of particle swarm optimization algorithms by addressing the problem of premature convergence. In the proposed local search technique, a potential particle position in the solution search space is collectively constructed by a number of randomly selected particles in the swarm. The number of times the selection is made varies with the dimension of the optimization problem and each selected particle donates the value in the location of its randomly selected dimension from its personal best. After constructing the potential particle position, some local search is done around its neighbourhood in comparison with the current swarm global best position. It is then used to replace the global best particle position if it is found to be better; otherwise no replacement is made. Using some well-studied benchmark problems with low and high dimensions, numerical simulations were used to validate the performance of the improved algorithms. Comparisons were made with four different PSO variants, two of the variants implement different local search technique while the other two do not. Results show that the improved algorithms could obtain better quality solution while demonstrating better convergence velocity and precision, stability, robustness, and global-local search ability than the competing variants.

## 1. Introduction

Optimization comes to focus when there are needs to plan, take decisions, operate and control systems, design models, or seek optimal solutions to varieties of problems faced from day to day by different people. A number of these problems, which can be formulated as continuous optimization problems, are often approached with limited resources. Dealing with such problems, most especially when they are large scale and complex, has attracted the development of different nature-inspired optimization algorithms. These algorithms display problem-solving capabilities for researchers to solve complex and challenging optimization problems with many success stories. Swarm-based techniques are a family of nature-inspired algorithms and are population-based in nature; they are also known as evolutionary computation techniques. Particle swarm optimization (PSO) technique is a member of swarm-based techniques which is capable of producing low cost, fast, and robust solutions to several complex optimization problems. It is a stochastic, self-adaptive, and problem-independent optimization technique and was originally proposed in 1995 by Eberhart and Kennedy as simulation of a flock of bird or the sociological behavior of a group of people [1, 2]. From the time this concept was brought into optimization, it has been used extensively in many fields which include function optimization and many difficult real-world optimization problems [3–5].

PSO technique was initially implemented with few lines of codes using basic mathematical operations with no major adjustment needed to adapt it to new problems and it was almost independent of the initialization of the swarm [6]. It needs few parameters to operate with for successful and efficient behavior in order to obtain quality solutions. To implement this technique, a number of particles, which are characterized by positions and velocities, called swarm are required to be randomly distributed in a solution search space depending on the boundaries defined for the design variables of the problem being optimized. The number of design variables determines the dimensionality of the search space. If $d$-dimensional space is considered, the position and velocity of each particle are represented as the vectors $X_i = (x_{i1}, \ldots, x_{id})$ and $V_i = (v_{i1}, \ldots, v_{id})$, respectively. Every particle has a memory of its personal experience which is communicated to all reachable neighbours in the search space to guide the direction of movement of the swarm. Also, the quality of each particle (solution) is determined by the objective function of the problem being optimized and the particle with best quality is taken as the global solution towards which other particles will converge. The common practice is for the technique to maintain a single swarm of particles throughout its operation. This process of seeking optimal solution involves the adjustments of the position and velocity of each particle in each iteration using

$$V_i(t+1) = \omega V_i(t) + \text{coeff}_1 (P_i - X_i) + \text{coeff}_2 (P_g - X_i), \quad (1)$$

$$X_i(t+1) = X(t) + V_i(t+1). \quad (2)$$

In (1), $P_i$ and $P_g$ are vectors representing the $i$th particle personal best and swarm global best positions, respectively; $\text{coeff}_1 = c_1 r_1$ and $\text{coeff}_2 = c_2 r_2$; $c_1$ and $c_2$ are acceleration factors known as cognitive and social scaling parameters that determine the magnitude of the random forces in the direction of $P_i$ and $P_g$; $r_1$ and $r_2$ are random numbers between 0 and 1; $t$ is iteration index. The symbol $\omega$ is the inertia weight parameter which was introduced into the original PSO in [7]. The purpose of its introduction was to help the PSO algorithm balance its global and local search activities.

There are possibilities of the positions and velocities of the particles in the swarm increasing in value beyond necessary when they are updated. As a measure, the positions are clamped in each dimension to the search range $[X_{\min}, X_{\max}]$ of the design variables, where $X_{\min}$ and $X_{\max}$ represent the lower and upper bounds of a particle's position, respectively, while their velocities are controlled to be within a specified range $[V_{\min}, V_{\max}]$, where $V_{\min}$ and $V_{\max}$ represent the lower and upper bounds of a particle's velocity, respectively. The idea of velocity clamping which was introduced by [1, 2, 8] and extensively experimented with in [9] has led to significant improvement as regards the performance of PSO. This is so because the particles could concentrate, taking reasonably sized steps to search through the search space rather than bouncing about excessively. A major feature that characterizes an efficient optimization algorithm is the ability to strike a balance between local and global search. Global search involves the particles being able to advance from a solution to other parts of the search space and locate other promising candidates while local search means that the particle is capable of exploiting the neighbourhood of the present solution for other promising candidates. In PSO, as the rate of information sharing increases among the particles they migrate towards the same direction and region in the search space. If any of the particles could not locate any better global solution after some time, they will eventually converge about the existing one which may not be the global minimum due to lack of exploration power; this is known as premature convergence. This type of behaviour is more likely when the swarm of particles is overconcentrated. It could also occur when the optimization problem is of high dimension and/or nonconvex. One of the possible ways to prevent this premature convergence is to embed a local search technique into PSO algorithm to help improve the quality of each solution by searching its neighbourhood. After the improvement, better information is communicated among the particles thereby increasing the algorithm's ability to locate better global solution in course of optimization. Hill climbing, modified Hooke and Jeeves, gradient descent, golden ratio, Stochastic local search, adaptive local search, local interpolation, simulated annealing, and chaotic local search are different local search techniques that have been combined with PSO to improve its local search ability [10–18].

In this paper, a different local search technique was proposed to harness the global search ability of PSO and improve on its local search efforts. This technique is based on the collective efforts of randomly selected (with replacement) particles a number of times equal to the size of the problem dimension. When a particle is selected, it is made to contribute the value in the position of its randomly selected dimension from its personal best. The contributed values are then used to form a potential global best solution which is further refined. This concept could offer PSO the ability to enhance its performance in terms of convergence speed, local search ability, robustness, and increased solution accuracy. The local search technique was hybridized with two of the existing PSO variants, namely, random inertia weight PSO (RIW-PSO) and linear decreasing inertia weight PSO (LDIW-PSO), to form two new variants. Numerical simulations were performed to validate the efficiencies of each of them and some statistical analyses were performed to ascertain any statistically significant difference in performance between the proposed variants and the old ones. From the results obtained it was shown that the proposed variants are very efficient.

In the sections that follow, RIW-PSO and LDIW-PSO are briefly described in Section 2; the motivation and description of the proposed local search technique are presented in Section 3 while the improved PSO with local search technique is described in Section 4. Numerical simulations are performed in Section 5 and Section 6 concludes the paper.

## 2. The Particle Swarm Optimization Variants Used

Two PSO variants were used to validate the proposed improvement of the performance of PSO technique. The variants are LDIW-PSO and RIW-PSO. These were chosen because of the evidence available in the literature that they are less efficient in optimizing many continuous optimization problems [19–21]. These variants are succinctly described below.

*2.1. PSO Based on Linear Decreasing Inertia Weight (LDIW-PSO).* This variant was proposed in [9] after the inertia weight parameter was introduced into the original PSO by [7]. It implements the linear decreasing inertia weight strategy represented in (3) which decreases from some high which facilitates exploration to a low value which on the other hand promotes exploitation. This greatly improved the performance of PSO. LDIW-PSO does global search at the beginning and converges quickly towards optimal positions but lacks exploitation power [9] and the ability required to jump out of the local minimum most especially when being in the multimodal landscape. Some improvements on LDIW-PSO exist in the literature [6, 9, 22]:

$$\omega_{\mathrm{i}} = \left(\omega_{\mathrm{start}} - \omega_{\mathrm{stop}}\right)\left(\frac{\mathrm{MAX}_{\mathrm{itr}} - \mathrm{i}}{\mathrm{MAX}_{\mathrm{itr}}}\right) + \omega_{\mathrm{stop}}, \quad (3)$$

where $\omega_{\mathrm{start}}$ and $\omega_{\mathrm{stop}}$ are the initial and final values of inertia weight, $i$ is the current iteration number, $\mathrm{MAX}_{\mathrm{itr}}$ is the maximum iteration number, and $\omega_{\mathrm{i}} \in [0,1]$ is the inertia weight value in the $i$th iteration. Apart from the problem of premature convergence, this variant was found inefficient in tracking a nonlinear dynamic system because of the difficulty in predicting whether exploration (a larger inertia weight value) or exploitation (a smaller inertia weight) will be better at any given time in the search space of the nonlinear dynamic system [23].

*2.2. PSO Based on Random Inertia Weight (RIW-PSO).* Due to the improved performance of PSO when the constant inertia weight was introduced into it [7], a new era of research was indirectly initiated and this has attracted the attentions of many researchers in the field. The inefficiency of linear decreasing inertia weight, which linearly decreases from 0.9 to 0.4, in tracking a nonlinear dynamic system prompted the introduction of RIW which randomly varies within the same range of values. Random adjustment is one of the strategies that have been proposed to determine the inertia weight value to further improve on the performance of PSO. This strategy is nonfeedback in nature and the inertia weight takes different value randomly at each iteration, from a specified interval. In line with this, random inertia weight strategy represented in (4) was introduced into PSO by [23] to enable the algorithm track and optimize dynamic systems. In the equation, rand() is a uniform random number in the interval [0, 1] which make the formula generate a number randomly varying between 0.5 and 1.0, with a mean value of 0.75. When $c_1$ and $c_2$ are set to 1.494, the algorithm seems to demonstrate better optimizing efficiency. The motivation behind the selection of these values was Clerc's constriction factor [23]:

$$\omega = 0.5 + \frac{\mathrm{rand}\,()}{2}. \quad (4)$$

Not much is recorded in the literature regarding the implementation of this variant of PSO. Some of the few implementations found in the literature are recorded in [6, 20–22].

## 3. Proposed Local Search Technique

The basic principle underlying the optimizing strategy of PSO technique is that each particle in the swarm communicates their discoveries to their neighbours and the particle with the best discovery attracts others. While this strategy looks very promising, there is the risk of the particles being susceptible to premature convergence, especially when the problem to be optimized is multimodal and high in dimensionality. The reason is that the more the particles share their discoveries among themselves, the higher their identical behaviour is until they converge to the same area in the solution search space. If none of the particle could discover better global best, after some time all the particles will converge about the existing global best which may not be the global minimizer.

One of the motivations for this local search technique is the challenge of premature convergence associated with PSO technique which affects its reliability and efficiency. Another motivation is the decision-making strategy used by the swarm in searching for optimal solution to optimization problems. The decision is dictated by a single particle in the swarm; that is, other particles follow the best particle among them to search for better solution. Involving more than one particle in the decision making could lead to a promising region in the search space where optimal solution could be obtained.

The description of the local search technique is as follows: after all the particles have obtained their various personal best positions, each particle has an equal chance of being selected to contribute its idea towards how a potential location in the search space where better global best could be obtained. As a result, a number of particles equal to the dimension of the problem being optimized are randomly selected (with replacement). Each selected particle contributes an idea by donating the value in the location of its randomly selected dimension from its personal best. All the ideas contributed by the selected particles are collectively used (hybridized) to construct a potential solution in the solution search space. After constructing the potential solution, some searches are locally done around its neighbourhood with the hope of locating a better solution in comparison with the current global solution. If a better solution is found, it is then used to replace the current global solution; otherwise no replacement is made.

In this local search, the potential new position is denoted by $\vec{y}$ and is sampled from the neighbourhood of the collectively constructed potential global solution represented as $\vec{P}$ by

$$\vec{y} \longleftarrow \vec{P} + \vec{a}, \quad (5)$$

where $\vec{a} \sim U[-\vec{r}, \vec{r}]$ is a random vector picked uniformly from the range $[-\vec{r}, \vec{r}]$ and $\vec{r}$ is the search radius which is initially set to max$R$ (maximum radius for local search). The local search technique moves from position $\vec{P}$ to position $\vec{y}$ when there is improvement to the fitness. If there is no improvement on the fitness of $\vec{P}$ by $\vec{y}$, the search radius is linearly decreased by multiplying it with a factor $q$ using

$$\vec{r} \longleftarrow q \times \vec{r},$$
$$q \longleftarrow (\max R - \min R) \times \frac{t}{\max T} + \min R, \quad (6)$$

```
$\vec{r}, \vec{a}, \vec{y} \leftarrow$ new arrays, each of length Dim
$\vec{r} \leftarrow \max R$
$t \leftarrow 0$
While $(t < \max T)$ do
    $t \leftarrow t + 1$
    $\vec{a} \leftarrow U(-\vec{r}, \vec{r})$
    for $j \leftarrow 1$ to problem Dimension
      randomly select any particle $i$
      randomly select a dimension $d$ from the personal best $P$ of the selected particle $i$
        $\vec{y}^{\,j} \leftarrow \vec{P}_i^{\,d} + \vec{a}^{\,j}$
    end for
     validate for search space boundary
    If $f(\vec{y}) < g\mathrm{Fit}$
      $\overrightarrow{g\mathrm{Pos}} \leftarrow \vec{y}$
      $g\mathrm{Fit} \leftarrow f(\vec{y})$
    else
      $q \leftarrow (\max R - \min R) \times \dfrac{t}{\max T} + \min R$
      $\vec{r} \leftarrow q \times \vec{r}$
    end if
end while
Return $\overrightarrow{g\mathrm{Pos}}$ and $g\mathrm{Fit}$
```

Algorithm 1: Collective local unimodal search.

```
Begin PSO$_{\mathrm{CLUS}}$ Algorithm
  Step 1. Definition Phase
        (1.1) function to optimize as $f$
        (1.2) Parameter
            (1.2.1) swarm size
            (1.2.2) problem dimension
            (1.2.3) solution search space
            (1.2.4) particle velocity range
  Step 2. Initialized phase
        For all particles randomly initialized in search space
        (2.1) position $x_i \leftarrow (x_{i1}, \ldots, x_{id})$
        (2.2) velocity $v_i \leftarrow (v_{i1}, \ldots, v_{id})$,
        (2.3) $p\mathrm{best}_i \leftarrow (x_{i1}, \ldots, x_{id})$
        (2.4) $g\mathrm{best} \leftarrow$ best of $p\mathrm{best}_i$
        (2.5) evaluate $f(x_i)$ using objective function of problem
  Step 3. Operation Phase
        Repeat until a stopping criterion is satisfied
        (3.1). Compute inertia weight using any inertia weight formula
        (3.2). For each particle $i$
            (3.2.1). update $v_i$ for particle using (1)
            (3.2.2). validate for velocity boundaries
            (3.2.3). update $x_i$ for particle using (2)
            (3.2.4). validate for position boundaries
            (3.2.5). If $f(x_i) < f(p\mathrm{best}_i)$ then $p\mathrm{best}_i \leftarrow x_i$
        (3.3). $g\mathrm{best} \leftarrow$ best of $p\mathrm{best}i$
        (3.4). Implement local search using CLUS in Algorithm 1
  Step 4. Solution Phase
        (4.1). $x^* \leftarrow g\mathrm{best}$
        (4.2). $f^* \leftarrow f(g\mathrm{best})$
        (4.3). Return $x^*$ and $f^*$
End PSO$_{\mathrm{CLUS}}$ Algorithm
```

Algorithm 2: Algorithm for PSO$_{\mathrm{CLUS}}$.

Table 1: Parameter settings for experiment.

| Parameter | $\omega_{\min}$ | $\omega_{\max}$ | $c_1 = c_2$ | $V_{\min}$ | $V_{\max}$ | min$R$ | max$R$ | max$T$ |
|---|---|---|---|---|---|---|---|---|
| Value | 0.9 | 0.4 | 1.494 | $0.05 * X_{\min}$ | $0.05 * X_{\max}$ | 0.01 | 2.0 | 100 |

where max$T$ is the maximum number of times the neighbourhood of $\overrightarrow{P}$ is to be sampled, $t$ is the current time the neighbourhood is being sampled, and min$R$ is the minimum radius for the local search.

This proposed local search technique has been named collective local unimodal search (CLUS) technique. It has some trace of similarity in operation with local unimodal sampling (LUS) technique [24]. But they are quite different in the sense that, while LUS randomly picks a potential solution from the entire population, CLUS constructs a potential solution using the collective efforts of a randomly selected number of particles from the swarm. Also, CLUS uses a linear method to decrease the search radius (step size) in the neighbourhood of the potential solution which is different from the method applied by LUS during optimization. The CLUS technique is presented in Algorithm 1. In the technique, $g$Fit and $\overrightarrow{g\text{Pos}}$ represent the current global fitness value and its corresponding position in the search space.

## 4. Improved PSO with Collective Unimodal Local Search (PSO$_{\text{CLUS}}$)

The RIW-PSO increases convergence in early iterations and does more of global search activities but soon gets stuck in local optima because of lack of local search ability. Also, LDIW-PSO does global search at earlier part of its iteration but lacks enough momentum to do local search as it gets towards its terminal point of execution. The aim of this paper is to make a general improvement on the performance of PSO which can be applied to any of its variants. To achieve this, the two PSO variants were hybridized with the proposed collective local unimodal search (CLUS) technique which takes advantage of their global search abilities to do some neighbourhood search for better results. The improved PSO algorithm is presented in Algorithm 2.

## 5. Numerical Simulations

In this section, the improved algorithm (PSO$_{\text{CLUS}}$) was implemented using the inertia weight strategy of RIW-PSO and the variant was labeled $R$-PSO$_{\text{CLUS}}$. It was also implemented using the inertia weight strategy of LDIW-PSO and the variant was labeled $L$-PSO$_{\text{CLUS}}$. The performances of $R$-PSO$_{\text{CLUS}}$ and $L$-PSO$_{\text{CLUS}}$ were experimentally tested against those of RIW-PSO and LDIW-PSO, respectively. The maximum number of iterations allowed was 1000 for problems with dimensions less than or equal to 10, 2000 for 20-dimensional problems, and 3000 for 30-dimensional problems. A swarm size of 20 was used in all the experiments and twenty-five independent runs were conducted to collect data for analysis. The termination criteria for all the algorithms were set to be as maximum number of iterations relative to the problems' dimensions. A run, in which an algorithm is able to satisfy the set success criteria (see Table 1) before or at the maximum iteration, is considered to be successful. To further prove the efficiency of the proposed local search technique, the proposed PSO variants were also compared with some existing PSO variants hybridized with different local search techniques. They are PSO with golden ratio local search [15] and PSO with local interpolation search [18]. A total of 6 different experiments were conducted.

(i) $R$-PSO$_{\text{CLUS}}$ was compared with PSO with golden ratio local search (GLSPSO);

(ii) $R$-PSO$_{\text{CLUS}}$ was compared with PSO with local interpolation search (PSOlis);

(iii) $R$-PSO$_{\text{CLUS}}$ was compared with RIW-PSO;

(iv) $L$-PSO$_{\text{CLUS}}$ was compared with PSO with golden ratio local search (GLSPSO);

(v) $L$-PSO$_{\text{CLUS}}$ was compared with PSO with local interpolation search (PSOlis);

(vi) $L$-PSO$_{\text{CLUS}}$ was compared with LDIW-PSO.

The application software was developed in Microsoft Visual C# programming language.

*5.1. Test Problems.* A total of 21 problems were used in the experiments. These problems have different degrees of complexity and multimodality which represents diverse landscapes enough to cover many of the problems which can arise in global optimization problems. Shown in Table 2 are the problems dimensions, optimal fitness values, and success thresholds. Presented in Table 3 are the definitions, characteristics (US: unimodal separable, UN: unimodal nonseparable, MS: multimodal separable, and MN: multimodal nonseparable), and search ranges of the problems. More details on the benchmark problems can be found in [22, 25–27].

*5.2. Parameter Setting.* The additional parameters that were set in the experiment are inertia weight threshold for LDIW-PSO ($\omega_{\min}$ and $\omega_{\max}$), acceleration coefficients ($c_1$ and $c_2$), velocity thresholds ($V_{\min}$ and $V_{\max}$), minimum radius (min$R$), and maximum radius (max$R$) for local search as well as the maximum number of neighbourhood sampling (max$T$) during the local search. The respective settings of these parameters are shown in Table 1. The parameters $r_1$ and $r_2$ were randomly generated using the uniform random number generator. The values of $\omega_{\min}$ and $\omega_{\max}$ were chosen for LDIW-PSO based on the experiments conducted in [9]; values for $c_1$ and $c_2$ were chosen for RIW-PSO based on the recommendation in [23] and it was also used for LDIW-PSO because it was discovered in course of the experiments in this paper that these values make LDIW-PSO perform better than

Table 2: Benchmark problems.

| Number | Problem | Dimensions | Optimal value | Success threshold |
|---|---|---|---|---|
| 1 | Ackley | 10, 20, 30 | 0 | $10^{-5}$ |
| 2 | Booth | 2 | 0 | $10^{-5}$ |
| 3 | Easom | 2 | −1 | −1 |
| 4 | Griewank | 10, 20, 30 | 0 | $10^{-5}$ |
| 5 | Dixon-Price | 10, 20, 30 | 0 | $10^{-5}$ |
| 6 | Levy | 10, 20, 30 | 0 | $10^{-5}$ |
| 7 | Michalewicz | 5 | −4.687 | −4.687 |
| 8 | Noisy Quartic | 10, 20, 30 | 0 | $10^{-5}$ |
| 9 | Noncontinous Rastrigin | 10, 20, 30 | 0 | 20 |
| 10 | Rastrigin | 10, 20, 30 | 0 | 20 |
| 11 | Rosenbrock | 10, 20, 30 | 0 | 20 |
| 12 | Rotated Ellipsoid | 10, 20, 30 | 0 | $10^{-5}$ |
| 13 | Salomon | 5 | 0 | $10^{-5}$ |
| 14 | Schaffer's f6 | 2 | 0 | $10^{-5}$ |
| 15 | Schwefel | 10, 20, 30 | | |
| 16 | Schwefel P2.22 | 10, 20, 30 | 0 | $10^{-5}$ |
| 17 | Shubert | 2 | −186.7309 | −186.7309 |
| 18 | Sphere | 10, 20, 30 | 0 | $10^{-5}$ |
| 19 | Step | 10, 20, 30 | 0 | $10^{-5}$ |
| 20 | Sum Squares | 10, 20, 30 | 0 | $10^{-5}$ |
| 21 | Trid | 6 | −50 | −50 |

the commonly used value of 2.0. The settings for $V_{\min}$ and $V_{\max}$ were done based on the outcome of experimental studies in [8].

*5.3. Performance Measurement.* The efficiency of the algorithms was tested against the set of benchmark problems given in Table 2 and numerical results obtained were analyzed using the criteria that are listed below. All the results are presented in Tables 4 to 20.

(i) Best fitness solution: the best of the fitness solution among the solutions obtained during the runs.

(ii) Mean best fitness solution: this is a measure of the precision (quality) of the result that the algorithm can get within given iterations in all the 25 runs.

(iii) Standard deviation (Std. Dev.) of mean best fitness solution over 25 runs: this measures the algorithm's stability and robustness.

(iv) Average number of iterations an algorithm was able to reach the success threshold.

(v) Success rate (SR) = (Number of successful runs/Total number of runs) × 100: this is the rate at which the success threshold is met during the independent number of runs and is a reflection of the global search ability and robustness of the algorithm.

Statistical analysis using the Wilcoxon signed rank nonparametric test with 0.05 level of significance [28, 29] was also performed using the numerical results obtained by the algorithms, while box plots were used to analyze their variability in obtaining fitness values in all the runs.

*5.4. Results and Discussions.* Results obtained from all the experiments are discussed in this subsection to show the overall performance of the various algorithms. Presented in Tables 4, 5, 6, 7, 8, and 9 are the numerical results obtained and used to compare $R$-PSO$_{\text{CLUS}}$ and $L$-PSO$_{\text{CLUS}}$ with GLSPSO. $R$-PSO$_{\text{CLUS}}$ and $L$-PSO$_{\text{CLUS}}$ were also compared with PSOlis using the results presented in Table 10. The results in Tables 11–18 were obtained for the scaled and nonscaled test problems listed in Table 3; the results were used to validate RIW-PSO, $R$-PSO$_{\text{CLUS}}$, LDIW-PSO, and $L$-PSO$_{\text{CLUS}}$. In each of the tables, for ease of observation, bold values represent the better results and "–" means that the algorithm could not satisfy the success threshold in any of the runs. The Wilcoxon sign rank nonparametric test, which is used as an alternative to the paired $t$-test when the results cannot be assumed to be normally distributed, was applied to test the statistical significance differences between RIW-PSO and $R$-PSO$_{\text{CLUS}}$ as well as LDIW-PSO and $L$-PSO$_{\text{CLUS}}$.

*5.4.1. Comparison of R-PSO$_{CLUS}$ and Golden Ratio Local Search Based PSO (GLSPSO).* The results in Tables 4–6 show the performance and abilities of $R$-PSO$_{\text{CLUS}}$ and GLSPSO optimizing the test problems over three different problem dimensions. The results of GLSPSO were obtained from [15]. A large problem space was used for all the problems to verify the superiority between the two different local search techniques hybridized with the PSO variants. From the results it is evident that $R$-PSO$_{\text{CLUS}}$ is superior to GLSPSO. Apart from *Ackley* problem (across the three dimensions) and *Rosenbrock* (in dimension 100), $R$-PSO$_{\text{CLUS}}$ outperformed GLSPSO. It was able to obtain optimal minimum for some of the problems, demonstrating better exploitation ability, convergence precision, and solution quality.

*5.4.2. Comparison between L-PSO$_{CLUS}$ and GLSPSO.* To further demonstrate the efficiency of the proposed local search technique, $L$-PSO$_{\text{CLUS}}$ was also implemented and results were compared with the results of GLSPSO obtained from [15]. Three different types of dimensions were also used for the problems. As can be observed in Tables 7–9, across the three dimensions, GLSPSO was only able to perform better than $L$-PSO$_{\text{CLUS}}$ in *Ackley* problem. Apart from *Griewank* problem (dimension 10), GLSPSO was outperformed by $L$-PSO$_{\text{CLUS}}$ in the remaining four problems. $L$-PSO$_{\text{CLUS}}$ was able to obtain global optimum for *Griewank* and *Sphere* problems across the three dimensions, but, for *Rastrigin*, it was able to get global minimum for dimension 10. Again, the proposed local search technique demonstrates better exploitation ability than GLSPSO.

*5.4.3. Comparison of R-PSO$_{CLUS}$ and L-PSO$_{CLUS}$ with PSOlis.* Presented in Table 10 is the result obtained by $L$-PSO$_{\text{CLUS}}$

Table 3: Benchmark problems.

| Number | Problem | Formulation | Feature | Search range |
|---|---|---|---|---|
| 1 | Ackley | $f(\overrightarrow{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e$ | MN | ±32 |
| 2 | Booth | $f(\overrightarrow{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$ | MN | ±10 |
| 3 | Easom | $f(\overrightarrow{x}) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1-\pi)^2 - (x_2-\pi)^2\right)$ | UN | ±100 |
| 4 | Griewank | $f(\overrightarrow{x}) = \frac{1}{4000}\left(\sum_{i=1}^{d} x_i^2\right) - \left(\prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1$ | MN | ±600 |
| 5 | Dixon-Price | $f(\overrightarrow{x}) = (x_1 - 1)^2 + \sum_{i=2}^{d} i\left(2x_i^2 - x_{i-1}\right)^2$ | UN | ±10 |
| 6 | Levy | $f(\overrightarrow{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{d-1}(y_i - 1)^2\left(1 + 10\sin^2(\pi y_i + 1)\right) + (y_d - 1)^2\left(1 + \sin^2(2\pi x_d)\right),$<br>where $y_i = 1 + \frac{x_i - 1}{4}$, and $i = 1, 2, \ldots, d$ | MN | ±10 |
| 7 | Michalewicz | $f(\overrightarrow{x}) = -\sum_{i=1}^{d}\sin(x_i)\left[\sin\left(\frac{ix_i^2}{\pi}\right)\right]^{2m}$, where $m = 10$ | MS | $[0, \pi]$ |
| 8 | Noisy Quartic | $f(\overrightarrow{x}) = \sum_{i=1}^{d} ix_i^4 + \text{random}(0, 1)$ | US | ±1.28 |
| 9 | Noncontinous Rastrigin | $f(\overrightarrow{x}) = \sum_{i=1}^{d}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right)$<br>$y_i = \left\{\begin{matrix} x_i & \text{if } \lvert x_i\rvert < 0.5 \\ \frac{\text{round}(2x_i)}{2} & \text{if } \lvert x_i\rvert \geq 0.5 \end{matrix}\right\}$ | MS | ±5.12 |
| 10 | Rastrigin | $f(\overrightarrow{x}) = \sum_{i=1}^{d}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ | MS | ±5.12 |
| 11 | Rosenbrock | $f(\overrightarrow{x}) = \sum_{i=1}^{d-1}\left(100\left(x_{i+1} - x_i^2\right)^2\right) + (x_i - 1)^2$ | UN | ±30 |
| 12 | Rotated Ellipsoid | $f(\overrightarrow{x}) = \sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$ | UN | ±100 |
| 13 | Salomon | $f(\overrightarrow{x}) = -\cos\left(2\pi\sum_{i=1}^{d} x_i^2\right) + 0.1\sqrt{\sum_{i=1}^{d} x_i^2} + 1$ | MN | ±100 |
| 14 | Schaffer's f6 | $f(\overrightarrow{x}) = \sum_{i=1}^{d-1}\left(0.5 + \frac{\sin^2\left(\sqrt{x_{i+1}^2 + x_i^2}\right) - 0.5}{\left(0.001\left(x_{i+1}^2 + x_i^2\right) + 1\right)^2}\right)$ | MN | ±100 |
| 15 | Schwefel | $f(\overrightarrow{x}) = \sum_{i=1}^{n} - x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$ | MS | ±500 |
| 16 | Schwefel P2.22 | $f(\overrightarrow{x}) = \sum_{i=1}^{d}\lvert x_i\rvert + \prod_{i=1}^{d}\lvert x_i\rvert$ | UN | ±10 |
| 17 | Shubert | $f(\overrightarrow{x}) = \prod_{i=1}^{d}\left(\sum_{j=1}^{5} j\cos\left((j+1)sx_i + j\right)\right)$ | MN | ±10 |
| 18 | Sphere | $f(\overrightarrow{x}) = \sum_{i=1}^{d} x_i^2$ | US | ±100 |
| 19 | Step | $f(\overrightarrow{x}) = \sum_{i=1}^{d}\left(\lfloor x_i + 0.5\rfloor\right)^2$ | US | ±10 |
| 20 | SumSquares | | US | ±10 |
| 21 | Trid | $f(\overrightarrow{x}) = \sum_{i=1}^{d}(x_i - 1)^2 - \sum_{i=2}^{d} x_i x_{i-1}$ | UN | $\pm d^2$ |

Table 4: Comparison between GLSPSO and $R$-PSO$_{\text{CLUS}}$ for problems with dimension of 10.

| Problem | Ackley | | Griewank | | Rastrigin | | Rosenbrock | | Sphere | |
|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | 0.0364 | **0.0000** | 4.2879$e$ − 04 | **0.0000e + 00** | 8.8062 | **0.0000** | 2.6188 | **0.0000** | 4.7832$e$ − 04 | **3.1461e − 43** |
| Mean fitness | **0.3413** | 17.1371 | 0.0041 | **0.0016** | 29.4936 | **0.0000** | 9.0025 | **1.9971** | 0.0142 | **0.0000** |
| Worst fitness | **1.2653** | 20.0888 | **0.0419** | 0.0791 | 50.4781 | **0.0000** | 18.9887 | **3.1444** | 0.0476 | **0.0000** |
| Std. Dev. | **0.2762** | 6.7543 | **0.0061** | 0.0111 | 10.4372 | **0.0000** | 0.034 | **0.7262** | 0.0123 | **0.0000** |

Table 5: Comparison between GLSPSO and $R$-PSO$_{\text{CLUS}}$ for problems with dimension of 30.

| Problem | Ackley | | Griewank | | Rastrigin | | Rosenbrock | | Sphere | |
|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | GLSPSO | R-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | **2.2784** | 20.3075 | 0.0897 | **0.0000** | 109.5946 | **13.9247** | 175.8785 | **22.7589** | 1.9123 | **0.0000** |
| Mean fitness | **2.8398** | 20.4778 | 0.1257 | **0.0000** | 185.5221 | **36.3715** | 218.4976 | **27.5147** | 2.7449 | **0.0000** |
| Worst fitness | **3.2952** | 20.5792 | 0.2074 | **0.0000** | 229.6229 | **72.6581** | 259.2466 | **76.7433** | 3.9559 | **0.0000** |
| Std. Dev. | 0.2273 | **0.0574** | 0.0274 | **0.0000** | 24.9829 | **16.4882** | 21.8027 | **9.9182** | 0.4840 | **0.0000** |

Table 6: Comparison between GLSPSO and $R$-PSO$_{\text{CLUS}}$ for problems with dimension of 100.

| Problem | Ackley | | Griewank | | Rastrigin | | Rosenbrock | | Sphere | |
|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ | GLSPSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | **3.5148** | 20.9666 | 0.3195 | **0.0022** | 792.004 | **293.5795** | **1378.0** | 1867.2669 | 23.0614 | **0.1970** |
| Mean fitness | **3.6709** | 21.0691 | 0.4242 | **0.0230** | 881.0822 | **688.0048** | **1602.0** | 24909.8486 | 27.2534 | **4.7232** |
| Worst fitness | **3.7664** | 21.1306 | 0.4992 | **0.0923** | 934.9773 | **848.9927** | **1763.0** | 95519.4585 | 29.1615 | **16.1174** |
| Std. Dev. | 0.0551 | **0.0316** | 0.0303 | **0.0255** | **35.2341** | 103.1854 | **90.2874** | 21083.5791 | **1.2253** | 4.2498 |

Table 7: Comparison between GLSPSO and $L$-PSO$_{\text{CLUS}}$ for problems with dimension of 10.

| Problem | Ackley | | Griewank | | Rastrigin | | Rosenbrock | | Sphere | |
|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | 0.0364 | **0.0000** | 4.2879$e$ − 04 | **0.0000e + 00** | 8.8062 | **0.0000** | 2.6188 | **0.0000** | 4.7832$e$ − 04 | **6.4151e − 76** |
| Mean fitness | **0.3413** | 18.2504 | **0.0041** | 0.0042 | 29.4936 | **0.0000** | 9.0025 | **1.0516** | 0.0142 | **0.0000** |
| Worst fitness | **1.2653** | 20.0771 | **0.0419** | 0.1008 | 50.4781 | **0.0000** | 18.9887 | **2.8033** | 0.0476 | **0.0000** |
| Std. Dev. | **0.2762** | 5.4640 | **0.0061** | 0.0186 | 10.4372 | **0.0000** | **0.034** | 0.6449 | 0.0123 | **0.0000** |

Table 8: Comparison between GLSPSO and $L$-PSO$_{\text{CLUS}}$ for problems with dimension of 30.

| Problem | Ackley | | Griewank | | Rastrigin | | Rosenbrock | | Sphere | |
|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | **2.2784** | 20.3184 | 0.0897 | **0.0000** | 109.5946 | **0.1444** | 175.8785 | **0.0000** | 1.9123 | **0.0000** |
| Mean fitness | **2.8398** | 20.4631 | 0.1257 | **0.0000** | 185.5221 | **18.7372** | 218.4976 | **25.1359** | 2.7449 | **0.0000** |
| Worst fitness | **3.2952** | 20.5734 | 0.2074 | **0.0000** | 229.6229 | **38.8433** | 259.2466 | **77.4444** | 3.9559 | **0.0000** |
| Std. Dev. | 0.2273 | **0.0615** | 0.0274 | **0.0000** | 24.9829 | **8.4570** | 21.8027 | **13.2536** | 0.4840 | **0.0000** |

Table 9: Comparison between GLSPSO and $L$-PSO$_{\text{CLUS}}$ for problems with dimension of 100.

| Problem | Ackley | | Griewank | | Rastrigin | | Rosenbrock | | Sphere | |
|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ | GLSPSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | **3.5148** | 20.2136 | 0.3195 | **0.0000** | 792.004 | **212.0416** | 1378.0 | **93.7390** | 23.0614 | **0.0000** |
| Mean fitness | **3.6709** | 21.0491 | 0.4242 | **0.0000** | 881.0822 | **366.6521** | 1602.0 | **107.2300** | 27.2534 | **0.0000** |
| Worst fitness | **3.7664** | 21.1152 | 0.4992 | **0.0000** | 934.9773 | **504.2204** | 1763.0 | **428.1758** | 29.1615 | **0.0000** |
| Std. Dev. | **0.0551** | 0.1254 | 0.0303 | **0.0000** | **35.2341** | 68.2009 | 90.2874 | **56.9231** | 1.2253 | **0.0000** |

Table 10: Comparison between PSOlis, $R$-PSO$_{\text{CLUS}}$ and $L$-PSO$_{\text{CLUS}}$.

| Problem | Algorithm | | |
|---|---|---|---|
| | PSOlis | $R$-PSO$_{\text{CLUS}}$ | $L$-PSO$_{\text{CLUS}}$ |
| Ackley | 4.081$e$ − 03 | **9.263e − 13** | **3.7135e − 15** |
| Griewank | 2.673$e$ − 02 | **6.921e − 03** | **2.945e − 07** |
| Rastrigin | 2.005 | **1.948e − 09** | **8.893e − 06** |
| Rosenbrock | 3.987 | **5.180e − 01** | **2.338e − 01** |
| Sphere | 6.137$e$ − 14 | **2.197e − 27** | **2.401e − 54** |

in comparison with result for PSOlis from [18]. Again, the outstanding performance of $L$-PSO$_{\text{CLUS}}$ over its competitor is evidence that the proposed local search technique is very efficient and capable of complementing the global search ability of PSO to obtain quality results by making it overcome premature convergence.

*5.4.4. Comparison between RIW-PSO and R-PSO$_{\text{CLUS}}$.* The results presented in Table 11 are for the nonscaled test problems as optimized by the two algorithms while those in Tables 12–14 are for the scaled problems with 10, 20, and 30 dimensions, respectively. In Table 19 are the results obtained using the Wilcoxon sign rank nonparametric test.

*(1) Results for the Nonscaled Problems.* For the 7 nonscaled problems, Table 11 shows that there are no performance differences between the two algorithms in optimizing *Booth*, *Easom*, *Shubert,* and *Trid* problems. For *Michalewicz*, *Schaffer's f6*, and *Salomon*, $R$-PSO$_{\text{CLUS}}$ obtained more quality solutions and demonstrated better global search ability than RIW-PSO. The convergence curves in Figures 1(c) and 1(d) show that $R$-PSO$_{\text{CLUS}}$ has faster and better convergence. However, the $P$ value (0.190) obtained from the Wilcoxon sign test shown in Table 19 revealed that there is no statistical difference in the performance between the two algorithms for the nonscaled problems. Also, the two algorithms have equal median fitness.

*(2) Results for 10-Dimensional Problems.* For the scaled problems with 10 dimensions, Table 12 clearly reveals great differences in performance between RIW-PSO and $R$-PSO$_{\text{CLUS}}$. The two algorithms successfully optimized *Rastrigin*, *Rosenbrock*, *Rotated Ellipsoid*, *Schwefel 2.22*, *Sphere,* and *Sum Squares* problems with equal success rate of 100%, but $R$-PSO$_{\text{CLUS}}$ obtained significantly better mean fitness and standard deviation with fewer number of iterations. $R$-PSO$_{\text{CLUS}}$ was able to obtain the minimum optima for both *Rastrigin* and *Step* problems. For the other problems, $R$-PSO$_{\text{CLUS}}$ clearly outperformed RIW-PSO in solution quality, convergence precision, global search ability, and robustness, though none of them could meet the success threshold in optimizing the *Noisy Quartic* problem. The $P$ value (0.001) obtained from the Wilcoxon sign test presented in Table 19 indicates that there is statistically significant difference in performance between the two algorithms with a large effect size of $r = 0.6$ in favour of $R$-PSO$_{\text{CLUS}}$. The median fitness is also an evidence of this.

*(3) Results for 20-Dimensional Problems.* The same set of experiments was performed using the same scaled problems but with their dimensions increased to 20, which also increased their complexities except *Griewank*. The numerical results in Table 13 also show that there are great differences in performance between RIW-PSO and $R$-PSO$_{\text{CLUS}}$. The two algorithms had equal success rate of 100% in optimizing *Rosenbrock*, *Schwefel 2.22*, *Sphere,* and *Sum Squares* problems with $R$-PSO$_{\text{CLUS}}$ obtaining significantly better mean fitness (except *Rosenbrock*), standard deviation, and fewer number of iterations. Out of the remaining 10 problems $R$-PSO$_{\text{CLUS}}$ outperformed RIW-PSO in 9 of them with better solution quality, convergence precision, global search ability, and robustness; it also had success rate of 100% in 6 of the problems compared with RIW-PSO and was able to obtain global minimum for *Griewank* and *Step* problems. The algorithms could not meet the success criteria in optimizing the *Dixon-Price*, *Noisy Quartic,* and *Schwefel* problems, but $R$-PSO$_{\text{CLUS}}$ still performed better than RIW-PSO. The $P$ value (0.023) from Wilcoxon sign test as shown in Table 19 also indicates that there is statistically significant difference in performance between the two algorithms with a medium effect size of $r = 0.43$ in favour of $R$-PSO$_{\text{CLUS}}$. The median fitness value of $R$-PSO$_{\text{CLUS}}$ is smaller than that of RIW-PSO.

*(4) Results for 30-Dimensional Problems.* Table 14 represents the experimental results obtained by the two algorithms using the same scaled problems but with their dimensions scaled to 30, which further increased their complexities except *Griewank*. The results further show the great differences in performance between RIW-PSO and $R$-PSO$_{\text{CLUS}}$. Out of the 14 problems $R$-PSO$_{\text{CLUS}}$ had 100% success rate in 7 of them (4 multimodal and 3 unimodal) while RIW-PSO could only have in 3 of them (all unimodal). The two algorithms

Table 11: Results for RIW-PSO and $R$-PSO$_{\text{CLUS}}$ for the 7 nonscaled benchmark problems.

| Problem | Booth | | Easom | | Michalewicz | | Schaffer's f6 | | Salomon | | Shubert | | Trid-6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | 0.0000*e* + 00 | 0.0000*e* + 00 | −1.0000*e* + 00 | −1.0000*e* + 00 | −3.3453*e* + 00 | **−4.4371e + 00** | 0.0000*e* + 00 | 0.0000*e* + 00 | 9.9833*e* − 02 | **7.3608e − 29** | −1.8673*e* + 02 | −1.8673*e* + 02 | −5.0000*e* + 00 | −5.0000*e* + 00 |
| Mean fitness | 0.0000*e* + 00 | 0.0000*e* + 00 | −1.0000*e* + 00 | −1.0000*e* + 00 | −2.6034*e* + 00 | **−4.1008e + 00** | 4.7052*e* − 03 | **1.1659e − 03** | 9.9833*e* − 02 | **7.0073e − 02** | −1.8673*e* + 02 | −1.8673*e* + 02 | −5.0000*e* + 00 | −5.0000*e* + 00 |
| Std. Dev. | 0.0000*e* + 00 | 0.0000*e* + 00 | 6.6613*e* − 16 | 6.6613*e* − 16 | 4.2719*e* − 01 | **1.7866e − 01** | 4.8183*e* − 03 | **3.1573e − 03** | **6.2063e − 18** | 4.4236*e* − 02 | **3.7706e − 14** | 4.0594*e* − 14 | 8.6628*e* − 14 | **7.0555e − 14** |
| Av. iteration | 39.12 | **37.92** | 55.0 | **45.48** | — | — | 133.83 | **109.95** | — | **923.40** | **71.8** | 107.16 | 114.40 | **110.20** |
| SR (%) | 100 | 100 | 100 | 100 | 0 | 0 | 48 | **88** | 0 | **20** | 100 | 100 | 100 | 100 |

Table 12: Results for RIW-PSO and $R$-PSO$_{\text{CLUS}}$ for the 14 scaled benchmark problems with dimension of 10.

| Problem | Ackley | | Griewank | | Dixon-Price | | Levy | | Noisy Quartic | | Noncontinuous Rastrigin | | Rastrigin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | 3.9968*e* − 15 | **4.4409e − 16** | 7.3960*e* − 03 | **0.0000e + 00** | 2.4652*e* − 31 | 2.4652*e* − 31 | 1.4997*e* − 32 | 1.4997*e* − 32 | 5.9308*e* − 04 | **2.7756e − 05** | 6.0002*e* + 00 | **0.0000e + 00** | 3.9767*e* + 00 | **0.0000e + 00** |
| Mean fitness | 1.0316*e* − 01 | **2.8599e − 15** | 6.6238*e* − 02 | **2.8539e − 03** | 6.1333*e* − 01 | 6.1333*e* − 01 | 4.1630*e* − 01 | **1.4997e − 32** | 4.6455*e* − 03 | **1.1590e − 04** | 1.1640*e* + 01 | **7.1054e − 17** | 1.2208*e* + 01 | **0.0000e + 00** |
| Std. Dev. | 5.0537*e* − 01 | **1.6573e − 15** | 3.3218*e* − 02 | **1.3981e − 02** | 1.8086*e* − 01 | 1.8086*e* − 01 | 7.5307*e* − 01 | **0.0000e + 00** | 3.2781*e* − 03 | **5.9530e − 05** | 4.3533*e* + 00 | **3.4809e − 16** | 4.4684*e* + 00 | **0.0000e + 00** |
| Av. iteration | 287.88 | **263.68** | — | **464.16** | 295.50 | **258.50** | 127.31 | **99.32** | — | — | 40.49 | **12.44** | 49.44 | **25.00** |
| SR (%) | 96 | **100** | 0 | **96** | 8 | 8 | 52 | **100** | 0 | 0 | 92 | **100** | 100 | 100 |
| Problem | Rosenbrock | | Rotated Ellipsoid | | Schwefel | | Schwefel 2.22 | | Sphere | | Step | | Sum Squares | |
| Algorithm | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | **4.6541e − 03** | 3.7243*e* − 01 | 2.5690*e* − 27 | **5.9210e − 29** | −3.2818*e* + 03 | **−4.1898e+ 03** | 1.8348*e* − 32 | **2.7555e − 34** | 7.3673*e* − 61 | **5.4496e − 61** | 0.0000*e* + 00 | 0.0000*e* + 00 | **8.1835e − 62** | 1.9089*e* − 61 |
| Mean fitness | 1.4459*e* + 00 | **7.1832e − 01** | 1.0457*e* − 21 | **3.5437e − 24** | −2.6199*e* + 03 | **−4.1384e+ 03** | 3.3382*e* − 28 | **1.2830e − 30** | 4.1760*e* − 53 | **9.2787e − 55** | 8.0000*e* − 01 | **0.0000e + 00** | 2.1909*e* − 53 | **9.3329e − 54** |
| Std. Dev. | 1.6362*e* + 00 | **2.3127e − 01** | 3.0984*e* − 21 | **1.1617e − 23** | 3.3350*e* + 02 | **9.7987e + 01** | 1.3406*e* − 27 | **4.6701e − 30** | 1.2467*e* − 52 | **3.6764−54** | 1.9183 + 00 | **0.0000e + 00** | 1.0052*e* − 52 | **3.3839e − 53** |
| Av. iteration | 89.28 | **35.16** | 426.60 | **337.16** | — | **877.69** | 268.84 | **222.04** | 180.12 | **158.08** | 88.72 | **15.68** | 143.52 | **123.48** |
| SR (%) | 100 | 100 | 100 | 100 | 0 | **52** | 100 | 100 | 100 | 100 | 72 | **100** | 100 | 100 |

TABLE 13: Results for RIW-PSO and $R$-PSO$_{\text{CLUS}}$ for the 14 scaled benchmark problems with dimension of 20.

| Problem | Ackley | | Griewank | | Dixon-Price | | Levy | | Noisy Quartic | | Noncontinuous Rastrigin | | Rastrigin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | 7.5495*e* – 15 | **3.9968e – 15** | 0.0000*e* + 00 | **0.0000e + 00** | 6.6667*e* – 01 | 6.6667*e* – 01 | 1.4997*e* – 32 | 1.4997*e* – 32 | 3.9758*e* – 03 | **8.2784e – 05** | 1.2000*e* + 01 | **1.6698e – 03** | 1.4000*e* + 01 | **9.8524e – 03** |
| Mean fitness | 2.6017*e* – 01 | **3.9968e – 15** | 2.7935*e* – 02 | **0.0000e + 00** | 6.6667*e* – 01 | 6.6667*e* – 01 | 7.8421*e* – 01 | **1.4997e – 32** | 1.0286*e* – 02 | **2.7771e – 04** | 3.0521*e* + 01 | **4.4448e + 00** | 2.8521*e* + 01 | **4.3599e + 00** |
| Std. Dev. | 6.0386*e* – 01 | **0.0000e + 00** | 1.8216*e* – 02 | **0.0000e + 00** | **2.4222e – 16** | 4.1352*e* – 08 | 8.0230*e* – 01 | **0.0000e + 00** | 5.0796*e* – 03 | **1.3566e – 04** | 1.0922*e* + 01 | **2.5668e + 00** | 9.5087*e* + 00 | **1.9684e + 00** |
| Av. iteration | 580.67 | **488.52** | 402.00 | **382.80** | — | — | 248.00 | **200.04** | — | — | 51.5 | **86.72** | 56.2 | 89.08 |
| SR (%) | 84 | **100** | 4 | **100** | 0 | 0 | 16 | **100** | 0 | 0 | 16 | **100** | 20 | **100** |
| Problem | Rosenbrock | | Rotated Ellipsoid | | Schwefel | | Schwefel 2.22 | | Sphere | | Step | | Sum Squares | |
| Algorithm | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | **8.2827e – 01** | 7.4672*e* + 00 | 8.7694*e* – 04 | **7.9922e – 09** | −5.9318*e* + 03 | **−8.1903e+ 03** | 2.4775*e* – 22 | **3.5223e – 26** | 3.1840*e* – 43 | **2.9531e – 44** | 0.0000*e* + 00 | **0.0000e + 00** | 4.8722*e* – 45 | **2.1277e – 46** |
| Mean fitness | **9.1608e + 00** | 1.0488*e* + 01 | **7.2146e – 06** | 1.9591*e* – 05 | −4.5638*e* + 03 | **−7.0464e+ 03** | 8.5083*e* – 17 | **4.9420e – 23** | 3.9759*e* – 37 | **8.4151e – 38** | 1.5600*e* + 00 | **0.0000e + 00** | 8.3393*e* – 39 | **1.4859e – 39** |
| Std. Dev. | 3.5710*e* + 00 | **8.1620e – 01** | **1.9200e – 05** | 7.2466*e* – 05 | 6.0867*e* + 02 | **8.4004e + 02** | 3.9686*e* – 16 | **1.2554e – 22** | 1.2218*e* – 36 | **3.0436e – 37** | 1.6020*e* + 00 | **0.0000e + 00** | 1.4280*e* – 38 | **4.9918e – 39** |
| Av. iteration | 263.40 | **133.72** | 1763.10 | **1636.74** | — | — | 575.92 | **418.88** | 354.00 | **309.04** | 646.57 | **19.36** | 310.72 | **268.12** |
| SR (%) | 100 | 100 | 84 | **92** | 0 | 0 | 100 | 100 | 100 | 100 | 28 | **100** | 100 | 100 |

Table 14: Results for RIW-PSO and $R$-PSO$_{\text{CLUS}}$ for the 14 scaled benchmark problems with dimension of 30.

| Problem | Ackley | | Griewank | | Dixon-Price | | Levy | | Noisy Quartic | | Noncontinuous Rastrigin | | Rastrigin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | 1.4655e − 14 | **3.9968e − 15** | 0.0000e + 00 | **0.0000e + 00** | 6.6667e − 01 | 6.6667e − 01 | 2.6858e − 01 | **1.4997e − 32** | 3.3454e − 03 | **1.0873e − 04** | 1.9001e + 01 | **5.7746e − 02** | 1.7895e + 01 | **0.0000e + 00** |
| Mean fitness | 5.3345e − 01 | **4.1389e − 15** | 1.1200e − 02 | **0.0000e + 00** | 6.6667e − 01 | 6.6716e − 01 | 1.8286e + 00 | **1.4997e − 32** | 1.1252e − 02 | **3.1438e − 04** | 4.3331e + 01 | **9.0806e + 00** | 3.2489e + 01 | **6.5810e + 00** |
| Std. Dev. | 8.1543e − 01 | **6.9619e − 16** | 1.4757e − 02 | **0.0000e + 00** | **3.2634e − 16** | 2.4146e − 03 | 1.1781e + 00 | **0.0000e + 00** | 4.3926e − 03 | **1.1754e − 04** | 1.8889e + 01 | **5.1000e + 00** | 1.0662e + 01 | **5.7209e + 00** |
| Av. iteration | 959.88 | **776.20** | 628.00 | **525.48** | — | — | — | **348.76** | — | — | 107 | **377.72** | 141.00 | **379.35** |
| SR (%) | 64 | **100** | 36 | **100** | 0 | 0 | 0 | **100** | 0 | 0 | 4 | **100** | 4 | **92** |
| Problem | Rosenbrock | | Rotated Ellipsoid | | Schwefel | | Schwefel 2.22 | | Sphere | | Step | | Sum Squares | |
| Algorithm | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ | RIW-PSO | $R$-PSO$_{\text{CLUS}}$ |
| Best fitness | **4.0555e − 02** | 1.8942e + 01 | 3.2599e − 04 | **1.7148e − 04** | −7.9654e + 03 | **−1.0692e+ 04** | 2.7819e − 17 | **1.0011e − 24** | 5.7947e − 39 | **5.0490e − 42** | 0.0000e + 00 | 0.0000e + 00 | 1.7310e − 36 | **1.2387e − 44** |
| Mean fitness | 2.3794e + 01 | **1.9930e + 01** | **5.5035e − 03** | 1.8327e − 02 | −6.7376e + 03 | **−9.1033e+ 03** | 4.1597e − 13 | **6.4368e − 23** | 4.1277e − 33 | **3.6673e − 36** | 3.7600e + 00 | **0.0000e + 00** | 3.9240e − 33 | **5.9219e − 39** |
| Std. Dev. | 1.9052e + 01 | **5.2276e − 01** | **8.1952e − 03** | 4.4826e − 02 | 8.7747e + 02 | **8.6528e + 02** | 1.5547e − 12 | **7.6462e − 23** | 1.1473e − 32 | **8.2055e − 36** | 2.3200e + 00 | **0.0000e + 00** | 1.3909e − 32 | **1.2917e − 38** |
| Av. iteration | 1523.36 | 2813.14 | — | — | — | — | 1139.64 | **658.68** | 595.52 | **505.80** | 2881.00 | **21.48** | 526.28 | **436.08** |
| SR (%) | 44 | **56** | 0 | 0 | 0 | 0 | 100 | 100 | 100 | 100 | 4 | **100** | 100 | 100 |

Table 15: Results for LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ for the 7 nonscaled benchmark problems.

| Problem | Booth | | Easom | | Michalewicz | | Schaffer's f6 | | Salomon | | Shubert | | Trid-6 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | 0.0000*e* + 00 | 0.0000*e* + 00 | −1.0000*e* + 00 | −1.0000*e* + 00 | −3.4792*e* + 00 | **−4.3762e+ 00** | 0.0000*e* + 00 | 0.0000*e* + 00 | 9.9833*e* − 02 | **3.3189e − 65** | −1.8673*e* + 02 | −1.8673*e* + 02 | −5.0000*e* + 01 | −5.0000*e* + 01 |
| Mean fitness | 0.0000*e* + 00 | 0.0000*e* + 00 | −1.0000*e* + 00 | −1.0000*e* + 00 | −2.6736*e* + 00 | **−4.1056e+ 00** | 1.5545*e* − 03 | **1.1659e − 03** | 9.9833*e* − 02 | **8.7853e − 02** | −1.8673*e* + 02 | −1.8673*e* + 02 | −50000*e* + 01 | −50000*e* + 01 |
| Std. Dev. | 0.0000*e* + 00 | 0.0000*e* + 00 | 6.6613*e* − 16 | 6.6613*e* − 16 | 3.6921*e* − 01 | **1.6196e − 01** | 3.5619*e* − 03 | **3.1573e − 03** | **7.3434e − 18** | 3.2442*e* − 02 | **3.9382e − 14** | 7.5559*e* − 05 | **7.8572e − 14** | 9.3154*e* − 14 |
| Av. iteration | **79.48** | 81.16 | 107.16 | **73.12** | — | — | 279.90 | **174.14** | — | **742** | **175.52** | 208.20 | 287.48 | **281.72** |
| SR (%) | 100 | 100 | 100 | 100 | 0 | 0 | 84 | **88** | 0 | **12** | **100** | 96 | 100 | 100 |

Table 16: Results for LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ for the 14 scaled benchmark problems with dimension of 10.

| Problem | Ackley | | Griewank | | Dixon-Price | | Levy | | Noisy Quartic | | Noncontinuous Rastrigin | | Rastrigin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | 3.9968*e* − 15 | **4.4409e − 16** | 2.9510*e* − 02 | **0.0000e + 00** | 6.6667*e* − 01 | 6.6667*e* − 01 | 1.4997*e* − 31 | **1.4997e − 32** | 3.4641 − 04 | **1.1331e − 05** | 4.0001*e* + 00 | **0.0000e + 00** | 2.9825*e* + 00 | **0.0000e + 00** |
| Mean fitness | 4.9916*e* − 15 | **2.4335e − 15** | 8.4720*e* − 02 | **1.4639e − 06** | 6.6667*e* − 01 | 6.6667*e* − 01 | 1.9892*e* − 01 | **1.4997e − 32** | 2.9606*e* − 03 | **9.7778e − 05** | 1.2200*e* + 01 | **4.0007e − 02** | 9.9019*e* + 00 | **0.0000e + 00** |
| Std. Dev. | **1.5952e − 15** | 1.7635*e* − 15 | 2.7646*e* − 02 | **7.1705e − 06** | 1.5401*e* − 10 | **1.2278e − 12** | 4.4094*e* − 01 | **0.0000e + 00** | 2.0277*e* − 03 | **6.3873e − 05** | 6.2484*e* + 00 | **1.9600e − 01** | 5.2190*e* + 00 | **0.0000e + 00** |
| Av. iteration | 500.92 | **486.84** | — | **703.83** | — | — | 309.58 | **281.36** | — | — | 40.86 | **12.88** | 45.13 | **26.24** |
| SR (%) | 100 | 100 | 0 | **96** | 0 | 0 | 76 | **100** | 0 | 0 | 88 | **100** | 96 | **100** |
| Problem | Rosenbrock | | Rotated Ellipsoid | | Schwefel | | Schwefel 2.22 | | Sphere | | Step | | Sum Squares | |
| Algorithm | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | **2.7701e − 03** | 1.5581*e* − 01 | 2.1928*e* − 44 | **1.2232e − 52** | −3.2028*e* + 03 | **−4.1898e+ 03** | **3.3962e − 48** | 2.3988*e* − 46 | 9.8201*e* − 110 | **3.2307e − 112** | 0.0000*e* + 00 | 0.0000*e* + 00 | 2.3361*e* − 110 | **1.6446e − 112** |
| Mean fitness | 9.3256*e* − 01 | **7.2655e − 01** | 5.8802*e* − 32 | **4.6946e − 47** | −2.5079*e* + 03 | **−4.1317e+ 03** | 8.0458*e* − 23 | **5.4599e − 29** | 3.1249*e* − 101 | **1.7383e − 106** | 1.6000*e* − 01 | **0.0000e + 00** | **1.6238e − 105** | 6.7305*e* − 105 |
| Std. Dev. | 1.3984*e* + 00 | **5.1516e − 01** | 2.8672*e* − 31 | **1.6736e − 46** | 4.0568*e* + 02 | **1.2489e + 02** | 3.9336*e* − 22 | **2.6683e − 28** | 1.2642*e* − 100 | **3.3735e − 106** | 7.8384*e* − 01 | **0.0000e + 00** | **5.1760e − 105** | 3.1425*e* − 104 |
| Av. iteration | 155.20 | **69.64** | 566.44 | **559.20** | — | **798.45** | 478.12 | **450.76** | 385.52 | **374.28** | 73.79 | **15.76** | 340.56 | **327.24** |
| SR (%) | 100 | 100 | 100 | 100 | 0 | **44** | 100 | 100 | 100 | 100 | 96 | **100** | 100 | 100 |

Table 17: Results for LDIW-PSO and $L$-PSO$_{\mathrm{CLUS}}$ for the 14 scaled benchmark problems with dimension of 20.

| Problem | Ackley | | Griewank | | Dixon-Price | | Levy | | Noisy Quartic | | Noncontinuous Rastrigin | | Rastrigin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ |
| Best fitness | 1.4655*e* − 14 | **3.9968e − 15** | 7.3960*e* − 03 | **0.0000e + 00** | 6.6667*e* − 01 | 6.6667*e* − 01 | 1.4997*e* − 32 | 1.4997*e* − 32 | 8.8724*e* − 04 | **8.2862e − 05** | 9.0003*e* + 00 | **5.4958e − 02** | 7.8533*e* + 00 | **0.0000e + 00** |
| Mean fitness | 2.7869*e* − 13 | **3.9968e − 15** | 3.5766*e* − 02 | **1.1969e − 05** | 6.6667*e* − 01 | 6.6667*e* − 01 | 9.4811*e* − 01 | **1.4997e − 32** | 6.2935*e* − 03 | **1.8753e − 04** | 2.9441*e* + 01 | **5.0733e + 00** | 2.2627*e* + 01 | **1.5954e + 00** |
| Std. Dev. | 1.1016*e* − 12 | **0.0000e + 00** | 2.3388*e* − 02 | **3.7730e − 05** | **9.4022e − 16** | 2.6746*e* − 08 | 1.0290*e* + 00 | **0.0000e + 00** | 2.8130*e* − 03 | **9.1506e − 05** | 1.1371*e* + 01 | **2.4613e + 00** | 7.0594*e* + 00 | **2.9944e + 00** |
| Av. iteration | 785.08 | **773.48** | — | **813.50** | — | — | 545.13 | **518.48** | — | — | 93 | **88.40** | 256.17 | **144.32** |
| SR (%) | 100 | 100 | 0 | **88** | 0 | 0 | 32 | **100** | 0 | 0 | 16 | **100** | 48 | **100** |
| Problem | Rosenbrock | | Rotated Ellipsoid | | Schwefel | | Schwefel 2.22 | | Sphere | | Step | | Sum Squares | |
| Algorithm | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\mathrm{CLUS}}$ |
| Best fitness | **8.5273e − 01** | 7.8279*e* + 00 | 5.7451*e* − 10 | **3.1410e − 20** | −5.5568*e* + 03 | **−79200e + 03** | 2.1114*e* − 17 | **4.5537e − 28** | 3.4892*e* − 59 | **4.4204e − 68** | 0.0000*e* + 00 | 0.0000*e* + 00 | 2.7411*e* − 60 | **3.2424e − 71** |
| Mean fitness | **8.8496e + 00** | 9.1318*e* + 00 | 6.1269*e* − 05 | **1.7093e − 15** | −4.5047*e* + 03 | **−68180e + 03** | 3.5766*e* − 09 | **1.6526e − 23** | 1.0393*e* − 45 | **1.6274e − 58** | 8.0000*e* − 02 | **0.0000e + 00** | 1.4838*e* − 48 | **1.5378e − 59** |
| Std. Dev. | 3.4651*e* + 00 | **9.4307e − 01** | 1.7066*e* − 04 | **4.8692e − 15** | **5.0603+02** | 5.1886*e* + 02 | 8.6693*e* − 09 | **3.1511e − 23** | 3.1193*e* − 45 | **5.5196e − 58** | 2.7129*e* − 01 | **0.0000e + 00** | 5.6876*e* − 48 | **7.4256e − 59** |
| Av. iteration | 594.96 | 419.12 | 1495.45 | **1254.52** | — | — | 820.56 | **722.76** | 639.88 | **636.08** | 147.52 | **19.88** | 595.80 | **593.52** |
| SR (%) | 100 | 100 | 88 | **100** | 0 | 0 | 100 | 100 | 100 | 100 | 92 | **100** | 100 | 100 |

Table 18: Results for LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ for the 14 scaled benchmark problems with dimension of 30.

| Problem | Ackley | | Griewank | | Dixon-Price | | Levy | | Noisy Quartic | | Noncontinuous Rastrigin | | Rastrigin | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | 3.6993*e* − 13 | **3.9968e − 15** | 2.2204*e* − 16 | **0.0000e + 00** | 6.6667*e* − 01 | 6.6667*e* − 01 | 3.4554 − 28 | **1.4997e − 32** | 3.2281*e* − 03 | **8.0745e − 05** | 2.0001*e* + 01 | **2.2356e + 00** | 1.8889*e* + 01 | **0.0000e + 00** |
| Mean fitness | 5.6071*e* − 11 | **3.9968e − 15** | 1.2289*e* − 02 | **0.0000e + 00** | 6.6667*e* − 01 | 6.6667*e* − 01 | 2.0125*e* + 00 | **1.4997e − 32** | 7.2161*e* − 03 | **2.3200e − 04** | 4.1361*e* + 01 | **1.2883e + 01** | 3.3921*e* + 01 | **8.3217e + 00** |
| Std. Dev. | 2.1743*e* − 10 | **0.0000e + 00** | 1.5738*e* − 02 | **0.0000e + 00** | 3.8233*e* − 12 | **3.1157e − 14** | 1.6464*e* + 00 | **0.0000e + 00** | 3.4677*e* − 03 | **1.0514e − 04** | 1.3494*e* + 01 | **5.1402e + 00** | 9.7583 + 00 | **7.6697e + 00** |
| Av. iteration | 1245.68 | **1194.40** | 1050.23 | 1060.44 | — | — | 928.00 | **865.44** | — | — | — | **658.48** | 518.00 | 718.21 |
| SR (%) | 100 | 100 | 52 | **100** | 0 | 0 | 4 | **100** | 0 | 0 | 0 | **92** | 4 | **96** |
| Problem | Rosenbrock | | Rotated Ellipsoid | | Schwefel | | Schwefel 2.22 | | Sphere | | Step | | Sum Squares | |
| Algorithm | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ | LDIW-PSO | $L$-PSO$_{\text{CLUS}}$ |
| Best fitness | **1.4208e − 01** | 1.4715*e* + 01 | 1.4959*e* − 04 | **3.8064e − 11** | −7.7298*e* + 03 | **−1.0232e+ 04** | 6.6416*e* − 10 | **1.2902e − 24** | 9.3155*e* − 41 | **1.4592e − 57** | 0.0000*e* + 00 | 0.0000*e* + 00 | 1.0873*e* − 41 | **1.2031e − 57** |
| Mean fitness | 1.8350*e* + 01 | **1.7789e + 01** | 4.5745*e* − 02 | **1.8073e − 06** | −6.5026*e* + 03 | **−8.6435e+ 03** | 1.0569*e* − 04 | **2.4455e − 22** | 1.2325*e* − 30 | **6.2108e − 47** | 7.2000*e* − 01 | **0.0000e + 00** | 1.6409*e* − 28 | **2.0496e − 48** |
| Std. Dev. | 1.1240*e* + 01 | **1.8867e + 00** | 1.3691*e* − 01 | **4.7189e − 06** | **7.1966e + 02** | 1.0203*e* + 03 | 4.0792*e* − 04 | **2.8267e − 22** | 3.7363*e* − 30 | **3.0419e − 46** | 8.7270*e* − 01 | **0.0000e + 00** | 5.9071*e* − 28 | **9.3904e − 48** |
| Av. iteration | **1920.22** | 2078.70 | — | **2524.35** | — | — | 1576.75 | **1120.20** | 1033.04 | **1028.12** | 408.92 | **22.52** | 968.72 | **952.28** |
| SR (%) | 92 | 92 | 0 | **92** | 0 | 0 | 80 | **100** | 100 | 100 | 52 | **100** | 100 | 100 |

Table 19: Wilcoxon signed rank test on mean fitness obtained by RIW-PSO and $R$-PSO$_{\text{CLUS}}$ for the test problems.

| Measurement | Scaled problems | | | Nonscaled problems |
|---|---|---|---|---|
| | Dim = 10 | Dim = 20 | Dim = 30 | |
| $R$-PSO$_{\text{CLUS}}$ < RIW-PSO | 13 | 11 | 12 | 3 |
| $R$-PSO$_{\text{CLUS}}$ > RIW-PSO | 0 | 2 | 2 | 0 |
| $R$-PSO$_{\text{CLUS}}$ = RIW-PSO | 1 | 1 | 0 | 4 |
| $z$ | −3.190 | −2.274 | −2.606 | −1.604 |
| $P$ value | 0.001 | 0.023 | 0.009 | 0.190 |
| $r$ | 0.600 | 0.430 | 0.490 | — |
| Median | | | | |
| RIW-PSO | 0.847 | 0.144 | 0.272 | −1.000 |
| $R$-PSO$_{\text{CLUS}}$ | 0.000 | 0.000 | 0.000 | −1.000 |

Table 20: Wilcoxon signed rank test on mean fitness obtained by LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ for the test problems.

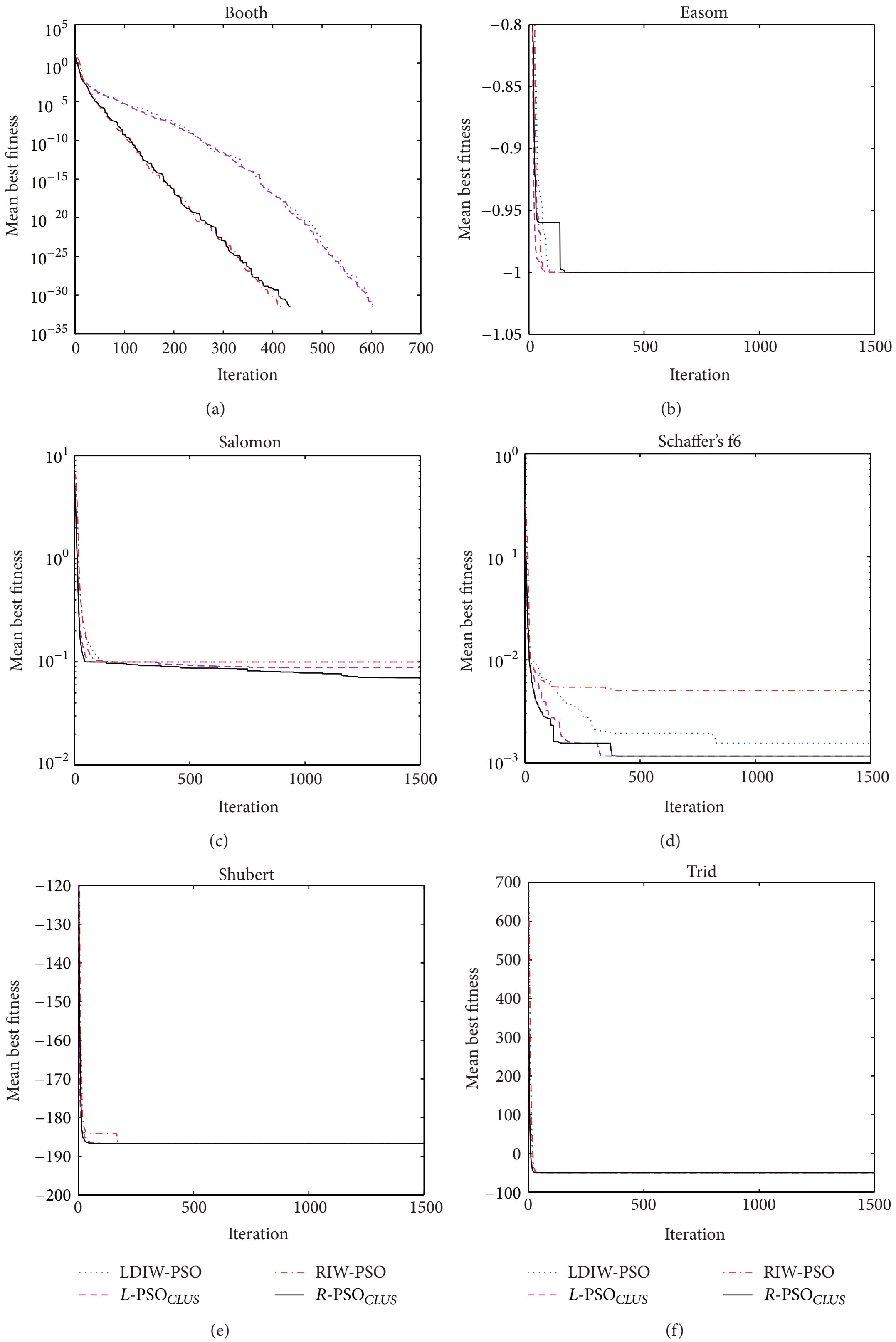
| Measurement | Scaled problems | | | Nonscaled |
|---|---|---|---|---|
| | Dim = 10 | Dim = 20 | Dim = 30 | |
| $L$-PSO$_{\text{CLUS}}$ < LDIW-PSO | 12 | 12 | 13 | 3 |
| $L$-PSO$_{\text{CLUS}}$ > LDIW-PSO | 1 | 1 | 0 | 0 |
| $L$-PSO$_{\text{CLUS}}$ = LDIW-PSO | 1 | 1 | 1 | 4 |
| $z$ | −2.988 | −2.552 | −3.181 | −1.604 |
| $P$ value | 0.003 | 0.011 | 0.001 | 0.190 |
| $r$ | 0.565 | 0.482 | 0.601 | — |
| Median | | | | |
| LDIW-PSO | 0.044 | 0.021 | 0.029 | −1.000 |
| $L$-PSO$_{\text{CLUS}}$ | 0.000 | 0.000 | 0.000 | −1.000 |

had equal success rate of 100% in optimizing *Schwefel 2.22*, *Sphere,* and *Sum Squares* problems with $R$-PSO$_{\text{CLUS}}$ obtaining significantly better mean fitness standard deviation and fewer number of iterations. Optimizing *Dixon-Price*, *Noisy Quartic*, *Rotated Ellipsoid,* and *Schwefel* problems, none of the algorithms could meet the success criteria, yet $R$-PSO$_{\text{CLUS}}$ still obtained better results than RIW-PSO. In all the 14 problems except *Rotated Ellipsoid*, $R$-PSO$_{\text{CLUS}}$ outperformed RIW-PSO and was able to obtain global minimum for *Griewank* and *Step* problems. The $P$ value (0.009) from Wilcoxon sign test in Table 19 is a confirmatory evidence that there is statistically significant difference in performance between RIW-PSO and $R$-PSO$_{\text{CLUS}}$ with a large effect size of $r = 0.49$ in favour of $R$-PSO$_{\text{CLUS}}$. The median fitness value of $R$-PSO$_{\text{CLUS}}$ is also smaller than that of RIW-PSO. The convergence graphs of six 30-dimensional test problems shown in Figure 2 demonstrate the speed and ability of convergence of the two algorithms. From the graphs it is clear that $R$-PSO$_{\text{CLUS}}$ demonstrates better convergence and global search ability than RIW-PSO. Besides it also possesses better ability to get out of local optima.

*5.4.5. Comparison between LDIW-PSO and L-PSO$_{\text{CLUS}}$.* Presented in Table 15 are the results for the nonscaled test problems as optimized by the two algorithms while those in Tables 16–18 are for the scaled problems with 10, 20, and 30 dimensions, respectively. The statistical analysis done by applying Wilcoxon sign rank nonparametric test is presented in Table 20.

*(1) Results for the Nonscaled Problems.* Results in Table 15 show that there are no clear performance differences between LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ in optimizing *Booth*, *Easom*, *Shubert,* and *Trid* problems; however, there are some not too significant differences in their average number of iterations to reach the success thresholds and standard deviation; in *Shubert*, LDIW-PSO obtained 100% success but $L$-PSO$_{\text{CLUS}}$ could not. Figures 1(a), 1(b), 1(e), and 1(f) show their convergence behaviour. Optimizing *Michalewicz*, *Schaffer's f6*, and *Salomon*, $L$-PSO$_{\text{CLUS}}$ obtained better quality solutions and has better search ability than LDIW-PSO. Also, the convergence graphs in Figures 1(c) and 1(d) show that $L$-PSO$_{\text{CLUS}}$ have faster and better convergence in *Schaffer's f6* and *Salomon*. The curves show that the two algorithms were trapped in local optima as shown by the flat parts of their curves and were able to escape from some of them. The $P$ value (0.190) in Table 20, obtained from the Wilcoxon sign test, indicates that there is no statistically significant difference between the two algorithms in performance.

*(2) Results for 10-Dimensional Problems.* Optimizing the 10-dimensional scaled problems, $L$-PSO$_{\text{CLUS}}$ had 100% success in 10 of the 14 problems (4 multimodal and 6 unimodal) while LDIW-PSO had 100% success in 6 problems (1 multimodal and 5 unimodal) as shown in Table 16. It is only $L$-PSO$_{\text{CLUS}}$



(a)

(b)

(c)

(d)

(e)

(f)

Figure 1: Convergence graphs for 6 of the nonscaled benchmark problems.

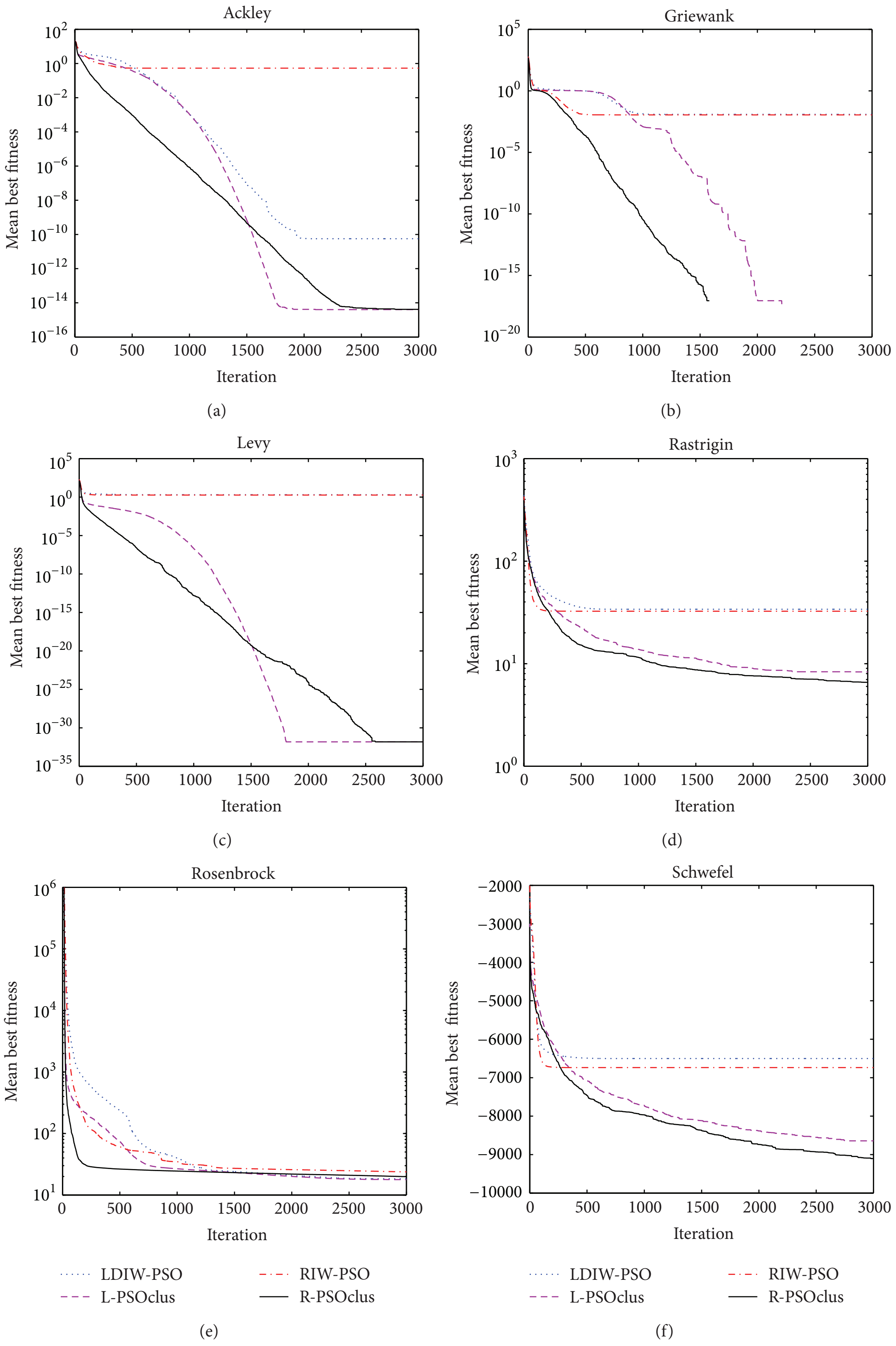


Figure 2: Convergence graphs for 6 of the scaled benchmark problems with dimension of 30.

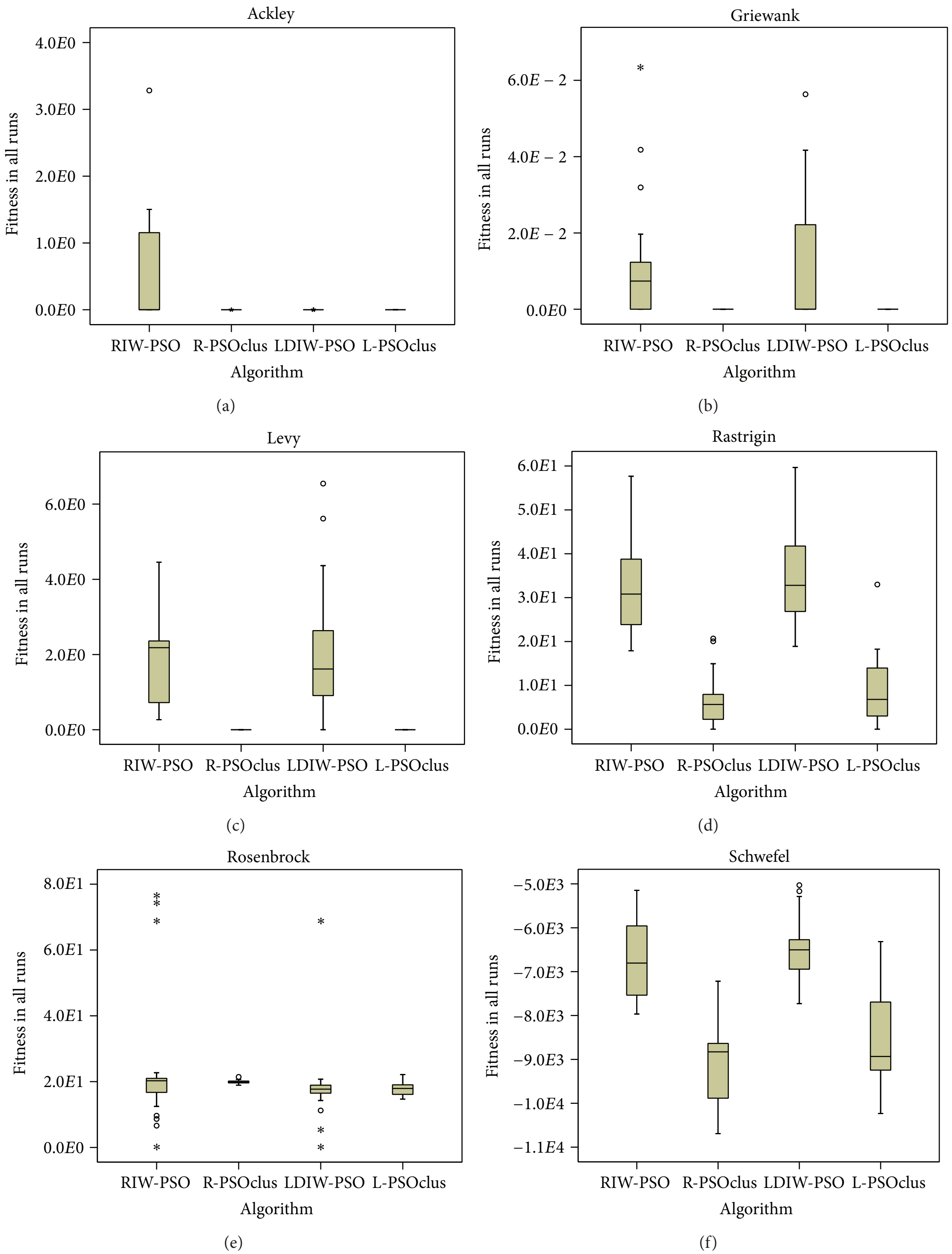


Figure 3: Box plots for 6 of the scaled test problems.

that could successfully obtain the minimum optima for both *Rastrigin* and *Step* problems but none could reach the success threshold for *Dixon-Price* and *Noisy Quartic*. In all the problems except *Dixon-Price* (where they have approximately equal performance) and *Sum Squares*, L-PSO$_{\text{CLUS}}$ clearly outperformed LDIW-PSO in obtaining better solution quality, convergence precision, global search ability, and robustness as well as fewer number of iterations. To confirm this, Wilcoxon sign test was performed on the mean best fitness over all the problems and results are presented in

Table 20; the $P$ value (0.003) obtained indicates that there is statistically significant difference in performance between the two algorithms with a large effect size of $r = 0.565$ in favour of $L$-PSO$_{\text{CLUS}}$ which also has a lower median value for the mean best fitness.

*(3) Results for 20-Dimensional Problems*. The numerical results in Table 17 also show that there are great differences in performance between LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ performing the same set of experiments but with the problems dimensions increased to 20. The two algorithms had equal success rate of 100% in optimizing *Ackley*, *Rosenbrock*, *Schwefel 2.22*, *Sphere,* and *Sum Squares* problems with $L$-PSO$_{\text{CLUS}}$ obtaining significantly better mean fitness (except *Rosenbrock*), standard deviation, and fewer number of iterations. $L$-PSO$_{\text{CLUS}}$ outperformed LDIW-PSO in 7 (5 multimodal and 2 unimodal) of the rest 9 problems and obtained better solution, convergence precision, global search ability, and robustness; it was also able to obtain global minimum for *Step* problem. The algorithms could not reach the success thresholds for *Dixon-Price*, *Noisy Quartic,* and *Schwefel* problems. The nonparametric test that was performed using Wilcoxon sign test, with results shown in Table 20, also confirms statistically significant difference in performance between the two algorithms with $P$ value (0.011) and a large effect size of $r = 0.482$ in the direction of $L$-PSO$_{\text{CLUS}}$. The median fitness value of $L$-PSO$_{\text{CLUS}}$ is also smaller than that of LDIW-PSO.

*(4) Results for 30-Dimensional Problems*. Scaling the dimensions of test problems to 30 to further increase their complexities, except *Griewank* which decreases in complexity with increased dimension, did not affect the better performance of $L$-PSO$_{\text{CLUS}}$ over LDIW-PSO. Presented in Table 18 are the experimental results obtained by the two algorithms optimizing the same scaled problems. The results indicate that there are great differences between LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ in performance. Out of the 14 problems $L$-PSO$_{\text{CLUS}}$ had 100% success rate in 6 of them (3 multimodal and 3 unimodal) while LDIW-PSO could only have in 3 (1 multimodal and 2 unimodal). They had equal success rate of 100% in optimizing *Ackley*, *Sphere,* and *Sum Squares* problems and 92% in *Rosenbrock* with $L$-PSO$_{\text{CLUS}}$ obtaining significantly better results. Optimizing *Dixon-Price*, *Noisy Quartic,* and *Schwefel* problems, none of the algorithms could reach the success threshold, yet $L$-PSO$_{\text{CLUS}}$ still obtained better results than LDIW-PSO, except in *Dixon-Price* where they had approximately the same performance. LDIW-PSO was not able to reach success threshold for *Noncontinuous Rastrigin* and *Rotated Ellipsoid* problems unlike $L$-PSO$_{\text{CLUS}}$. In all the 14 problems $L$-PSO$_{\text{CLUS}}$ conceded in none to LDIW-PSO and it was able to obtain global minimum for *Griewank* and *Step* problems. The $P$ value (0.001) in Table 20 further confirms that there is statistically significant difference between LDIW-PSO and $L$-PSO$_{\text{CLUS}}$ with a large effect size of $r = 0.601$ in the direction of $L$-PSO$_{\text{CLUS}}$. The median value for the mean fitness of $L$-PSO$_{\text{CLUS}}$ is also smaller than that of RIW-PSO. Figure 1 shows the convergence graphs of the two algorithms. From the graphs it is clear that $L$-PSO$_{\text{CLUS}}$ demonstrates better convergence speed, better ability to escape premature convergence, and global search ability than LDIW-PSO.

*5.4.6. Box Plots Analysis.* Other than using statistical test to observe the performance of RIW-PSO, $R$-PSO$_{\text{CLUS}}$, LDIW-PSO, and $L$-PSO$_{\text{CLUS}}$, box plots analysis was also performed for 6 of the scaled test problems with 30 dimensions; the results are presented in Figures 3(a)–3(f). Box plots give a direct visual comparison of both location and the dispersion of data. The four algorithms are plotted together to optimize space. In each of the plot, RIW-PSO is compared with $R$-PSO$_{\text{CLUS}}$ while LDIW-PSO is compared with $L$-PSO$_{\text{CLUS}}$. The plots strengthen and justify the better performance of PSO when used with the proposed local search technique.

## 6. Conclusion

A new local search technique has been proposed in this paper with the goal of addressing the problem of premature convergence associated with particle swarm optimization algorithms. The proposed local search was used to efficiently improve the performance of two existing PSO variants, RIW-PSO and LDIW-PSO. These variants have been known to be less efficient optimizing continuous optimization problems. In this paper they were hybridized with the local search to form two other variants $R$-PSO$_{\text{CLUS}}$ and $L$-PSO$_{\text{CLUS}}$. Some well-studied benchmark problems with low and high dimensions were used to extensively validate the performance of these new variants and comparisons were made with RIW-PSO and LDIW-PSO. They were also compared with two other PSO variants in the literature, which are hybridized with different local search techniques. The experimental results obtained show that the proposed variants successfully obtain better results with high quality while demonstrating better convergence velocity and precision, stability, robustness, and global-local search ability than the competing variants. This therefore shows that the local search technique proposed can help PSO algorithms execute effective exploitation in the search space to obtain high quality results for complex continuous optimization problems. This local search technique can be used with any population-based optimization algorithms to obtain quality solutions to simple and complex optimization problem.

Further study is needed on the parameter tuning of the proposed local search technique. Empirical investigation of the behaviour of the technique in optimizing problems with noise needs further study. The scalability of the algorithms for problems with higher dimension greater than 100 is essential. Finally, the proposed algorithm can be applied to real-world optimization problems.

## Conflict of Interests

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

## Acknowledgment

The authors acknowledge College of Agriculture, Engineering and Sciences, University of Kwazulu-Natal, South Africa, for their support towards this work.

## References

[1] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39–43, Nagoya, Japan, October 1995.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.

[3] V. N. Dieu, P. Schegner, and W. Ongsakul, "A newly improved particle swarm optimization for economic dispatch with valve point loading effects," in *Proceedings of the IEEE Power and Energy Society General Meeting*, pp. 1–8, July 2011.

[4] J. Bai, X. Zhang, and Y. Guo, "Different inertia weight PSO algorithm optimizing SVM kernel parameters applied in a speech recognition system," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '09)*, pp. 4754–4759, August 2009.

[5] M. M. Mansour, S. F. Mekhamer, and N. E.-S. El-Kharbawe, "A modified particle swarm optimizer for the coordination of directional overcurrent relays," *IEEE Transactions on Power Delivery*, vol. 22, no. 3, pp. 1400–1410, 2007.

[6] A. M. Arasomwan and A. O. Adewumi, "An adaptive velocity particle swarm optimization for high-dimensional function optimization," in *Proceedings of the IEEE Congress Evolutionary Computation (CEC '13)*, pp. 2352–2359, 2013.

[7] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, Anchorage, Alaska, USA, May 1998.

[8] G. I. Evers, *An automatic regrouping mechanism to deal with stagnation in particle swarm optimization [M.S. thesis]*, Graduate School of the University of Texas-Pan American, 2009.

[9] Y. Shi and R. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the 7th International Conference on Evolutionary Programming (EP '98)*, vol. 1447, pp. 591–600, San Diego, Calif, USA, March 1998.

[10] R. Akbari and K. Ziarati, "Combination of particle swarm optimization and stochastic local search for multimodal function optimization," in *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA '08)*, vol. 2, pp. 388–392, December 2008.

[11] C. Junying, Q. Zheng, L. Yu, and L. Jiang, "Particle swarm optimization with local search," in *Proceedings of the International Conference on Neural Networks and Brain (ICNNB '05)*, pp. 481–484, October 2005.

[12] B. Liu, L. Wang, Y.-H. Jin, F. Tang, and D.-X. Huang, "Improved particle swarm optimization combined with chaos," *Chaos, Solitons and Fractals*, vol. 25, no. 5, pp. 1261–1271, 2005.

[13] A. A. Mousa, M. A. El-Shorbagy, and W. F. Abd-El-Wahed, "Local search based hybrid particle swarm optimization algorithm for multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 3, pp. 1–14, 2012.

[14] H. Pan, X. Han, and M. Zheng, "Particle swarm-simulated annealing fusion algorithm and its application in function optimization," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, vol. 21, pp. 78–81, December 2008.

[15] Y. Sun, B. J. Wyk, and Z. Wang, "A new golden ratio local search based particle swarm optimization," in *Proceedings of the International Conference on Systems and Informatics (ICSAI '12)*, pp. 754–757, 2012.

[16] J. Tang and X. Zhao, "Particle swarm optimization using adaptive local search," in *Proceedings of the International Conference on Future BioMedical Information Engineering (FBIE '09)*, pp. 300–303, December 2009.

[17] Y.-J. Wang, "Improving particle swarm optimization performance with local search for high-dimensional function optimization," *Optimization Methods and Software*, vol. 25, no. 5, pp. 781–795, 2010.

[18] W.-B. Zhang, J.-Y. Chen, and Y.-Q. Ye, "Study on particle swarm optimization algorithm with local interpolation search," in *Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR '10)*, vol. 1, pp. 345–348, March 2010.

[19] A. M. Arasomwan and A. O. Adewumi, "On the performance of linear decreasing inertia weight particle swarm optimization for global optimization," *The Science World Journal*, vol. 2013, Article ID 860289, 2013.

[20] J. Ememipour, M. M. S. Nejad, M. M. Ebadzadeh, and J. Rezanejad, "Introduce a new inertia weight for particle swarm optimization," in *Proceedings of the IEEE 4th International Conference on Computer Sciences and Convergence Information Technology (ICCIT '09)*, pp. 1650–1653, November 2009.

[21] Y. Feng, G.-F. Teng, A.-X. Wang, and Y.-M. Yao, "Chaotic inertia weight in particle swarm optimization," in *Proceedings of the 2nd International Conference on Innovative Computing, Information and Control (ICICIC '07)*, September 2007.

[22] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing Journal*, vol. 11, no. 4, pp. 3658–3670, 2011.

[23] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the of the Congress on Evolutionary Computation*, vol. 1, pp. 94–100, Seoul, Korea, May 2001.

[24] M. E. H. Pedersen, *Tuning & simplifying heuristical optimization [Ph.D. thesis]*, School of Engineering Sciences, University of Southampton, Southampton, UK, 2010.

[25] S. Chetty and A. O. Adewumi, "Three new stochastic local search algorithms for continuous optimization problems," *Computational Optimization and Applications*, vol. 56, no. 3, pp. 675–721, 2013.

[26] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[27] B. A. Sawyerr, M. M. Ali, and A. O. Adewumi, "A comparative study of some real-coded genetic algorithms for unconstrained global optimization," *Optimization Methods and Software*, vol. 26, no. 6, pp. 945–970, 2011.

[28] C. Dytham, *Choosing and Using Statistics: A Biologist's Guide*, Wiley-blackwell, Malaysia, 3rd edition, 2011.

[29] J. Pallant, *SPSS Survival Manual*, McGraw-Hill, Singapore, 4th edition, 2010.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 748141, 11 pages
http://dx.doi.org/10.1155/2014/748141

*Research Article*

# A Novel Algorithm Combining Finite State Method and Genetic Algorithm for Solving Crude Oil Scheduling Problem

**Qian-Qian Duan, Gen-Ke Yang, and Chang-Chun Pan**

*Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Ministry of Education of China, Shanghai 200240, China*

Correspondence should be addressed to Gen-Ke Yang; sjtu1019@163.com and Chang-Chun Pan; 390635304@qq.com

Received 4 October 2013; Accepted 4 December 2013; Published 18 February 2014

Academic Editors: Q. Cheng and J. Yang

A hybrid optimization algorithm combining finite state method (FSM) and genetic algorithm (GA) is proposed to solve the crude oil scheduling problem. The FSM and GA are combined to take the advantage of each method and compensate deficiencies of individual methods. In the proposed algorithm, the finite state method makes up for the weakness of GA which is poor at local searching ability. The heuristic returned by the FSM can guide the GA algorithm towards good solutions. The idea behind this is that we can generate promising substructure or partial solution by using FSM. Furthermore, the FSM can guarantee that the entire solution space is uniformly covered. Therefore, the combination of the two algorithms has better global performance than the existing GA or FSM which is operated individually. Finally, a real-life crude oil scheduling problem from the literature is used for conducting simulation. The experimental results validate that the proposed method outperforms the state-of-art GA method.

## 1. Introduction

In recent years refineries have to explore all potential cost-saving strategies due to intense competition arising from fluctuating product demands and ever-changing crude prices. Scheduling of crude oil operations is a critical task in the overall refinery operations [1–3]. Basically, the optimization of crude oil scheduling operations consists of three parts [4]. The first part involves the crude oil unloading, mixing, transferring, and multilevel crude oil inventory control process. The second part deals with fractionation, reaction scheduling, and a variety of intermediate product tanks control. The third part involves the finished product blending and distributing process. In this paper, we focus on the first part, as it is a critical component for refinery scheduling operations. Scheduling of crude oil problem is often formulated as mixed integer nonlinear programming (MINLP) models [2, 5, 6]. The solution approaches for solving MINLP can be roughly divided into two categories [7]: deterministic approaches and stochastic approaches. Some deterministic methods have been available for many years [8]. These methods require the prior step of identification and elimination of nonconvexity and decompose the MINLP models into relevant nonlinear programming (NLP) and mixed integer linear programming (MILP) and then these subproblems have to be iteratively solved. The most common algorithms are branch and bound [9], outer-approximation [10], generalized benders decomposition [11], and so forth. Also, some commercial MINLP solvers have been developed for solving the problem at hand optimally [12]. However, the commercial solver can only handle MINLPs with special properties. The other stream of global optimization is the stochastic algorithms, for example, simulated annealing (SA), GA, and their variants [7]. GA proposed by Holland [13], because of their simple concept, easy scheme, and the global search capability independent of gradient information, have been developed rapidly. Much other attention is given to the development of GA for MINLP. For instance, Yokota et al. developed a penalty function that is suitable for solving MINLP problems [14]. Costa and Oliveira also implemented another type of penalty function to solve various MINLP problems, including industrial-scale problems [15]. They also noted that the evolutionary approach is efficient, in terms of the number of function evaluations, and is very suitable to handle the difficulties of the nonconvexity. Going one step further, some mixed coding methods were proposed, which include mixed-coding genetic algorithm [15] and information-guided genetic algorithm (IGA). Ponce-Ortega et al. [16] proposed a two-level approach based on GA to optimize the heat exchanger networks (HENs). The outer level is used to perform the structural optimization, for which a binary GA is used. Björk and Nordman [17] showed that the GA is very suitable to solve a large-scale heat exchanger network.

Obviously, the two different approaches previously discussed have their own advantages and disadvantages. On the one hand, a deterministic approach usually involves considerable algebra and undeviating analysis to the problem itself, whereas the evolutionary approach does not have this property. On the other hand, some deterministic approaches, such as mathematical programming, usually cannot provide practical solutions in reasonable time, whereas the evolutionary approach can generate satisfying solutions. In this work, a novel genetic algorithm which combined the finite state method and GA is proposed to solve crude oil scheduling problem. A MINLP model is formulated based on the single-operation sequencing (SOS) time representation. A deterministic finite automation (DFA) model which captures valid possible schedule sequences is constructed based on the sequencing rules. The initialization and mutation operation of GA is based on the model which builds legal schedules complying with sequencing rules and operation condition. Thus, the search space of the algorithm is substantially reduced as only legal sequence is explored. The rest of the paper is organized as follows: the MINLP model is specified in Section 2. Section 3 reviews the background of finite state theory. In Section 4, a novel genetic algorithm which combined the finite state method and GA is proposed to solve the MINLP model. A test problem is studied to verify our approach in Section 5. In the last section, conclusive remarks are given.

## 2. Mathematic Model

In this section, the MINLP model of refinery crude oil scheduling problem is described [18]. This problem has been widely studied from the optimization viewpoint since the work of Lee et al. [19]. It consists of crude oil unloading from marine vessels to storage tanks, transfer and blending between tanks, and distillation of crude mixtures. The goal is to maximize profit and meet distillation demands for each type of crude blend (e.g., low sulfur or high sulfur blends), while satisfying unloading and transfer logistics constraints, inventory capacity limitations, and property specifications for each blend. The logistics constraints involve nonoverlapping constraints between crude oil transfer operations.

*2.1. Sets.* The following sets will be used in the model.

(i) $T = \{1, \ldots, n\}$ is the set of priority-slots;

(ii) $W$ is the set of all operations: $W \triangleq W_U \cup W_T \cup W_D$;

(iii) $W_U \subset W$ is the set of unloading operations;

(iv) $W_T \subset W$ is the set of tank-to-tank transfer operations;

(v) $W_D \subset W$ is the set of distillation operations;

(vi) $R$ is the set of all operations: $R = R_V \cup R_S \cup R_C \cup R_D$;

(vii) $R_V \subset R$ is the set of vessels;

(viii) $R_S \subset R$ is the set of storage tanks;

(ix) $R_C \subset R$ is the set of charging tanks;

(x) $R_D \subset R$ is the set of distillation units;

(xi) $I_r \subset W$ is the set of inlet transfer operations on resource $r$;

(xii) $O_r \subset W$ is the set of outlet transfer operations on resource $r$;

(xiii) $C$ is the set of products (i.e., crudes);

(xiv) $K$ is the set of product properties (e.g., crude sulfur concentration).

*2.2. Parameters.* Parameters used in the paper are defined below:

(i) $H$ is the scheduling horizon;

(ii) $[\underline{V}_v^t, \overline{V}_v^t]$ are bounds on the total volume transferred during transfer operation $V$; in all instances, $\underline{V}_v^t = 0$ for all operations except unloading for which $\underline{V}_v^t = \overline{V}_v^t$ is the volume of crude in the marine vessel;

(iii) $[\underline{N_D}, \overline{N_D}]$ are the bounds on the number of distillations;

(iv) $[\underline{FR_v}, \overline{FR_v}]$ are flow rate limitations for transfer operation $v$;

(v) $S_r$ is the arrival time of vessel $r$;

(vi) $[\underline{x_{vk}}, \overline{x_{vk}}]$ are the limits of property $k$ of the blended products transferred during operation $v$;

(vii) $x_{ck}$ is the value of the property $k$ of crude $c$;

(viii) $[\underline{L}_r^t, \overline{L}_r^t]$ are the capacity limits of tank $r$;

(ix) $[\underline{D_r}, \overline{D_r}]$ are the bounds of the demand on products to be transferred out of the charging tank $r$ during the scheduling horizon;

(x) $G_c$ is the gross margin of crude $c$.

*2.3. Variables*

*2.3.1. Assignment Variables*

$$Z_{iv} \in \{0, 1\}, \quad i \in T, \ v \in W. \tag{1}$$

$Z_{iv} = 1$ if operation $v$ is assigned to priority-slot $i$; $Z_{iv} = 0$ otherwise.

*2.3.2. Time Variables*

$$S_{iv} \geq 0, \quad D_{iv} \geq 0, \quad i \in T,\ v \in W. \tag{2}$$

$S_{iv}$ is the start time of operation $v$ if it is assigned to priority slot $i$; $S_{iv} = 0$ otherwise.

$D_{iv}$ is the duration of operation $v$ if it is assigned to priority slot $i$; $D_{iv} = 0$ otherwise.

*2.3.3. Operation Variables*

$$V_{iv}^{t} \geq 0, \quad V_{ivc} \geq 0, \quad i \in T,\ v \in W,\ c \in C. \tag{3}$$

$V_{iv}^{t}$ is the total volume of crude transferred during operation $v$ if it is assigned to priority slot $i$; $V_{iv}^{t} = 0$ otherwise.

$V_{ivc}$ is the volume of crude $c$ transferred during operation $v$ if it is assigned to priority slot $i$; $V_{ivc} = 0$ otherwise.

*2.3.4. Resource Variables*

$$L_{ir}^{t}, L_{irc}, \quad i \in T,\ r \in R,\ c \in C. \tag{4}$$

$L_{ir}^{t}$ is the total accumulated level of crude in tank $r \in R_S \cup R_C$ before the operation was assigned to priority-slot $i$.

$L_{irc}$ is the accumulated level of crude $c$ in tank $r \in R_S \cup R_C$ before the operation was assigned to priority-slot $i$.

*2.4. Objective Function.* The objective is to maximize the gross margins of the distilled crude blends. Let $G_c$ be the individual gross margin of crude $c$,

$$\max \sum_{i \in T} \sum_{r \in R_D} \sum_{v \in I_r} \sum_{c \in C} G_c \cdot V_{ivc}. \tag{5}$$

*2.5. General Constraints.* It should be noted that the crude composition of blends in tanks is tracked instead of their properties. The distillation specifications are later enforced by calculating a posteriori the properties of the blend in terms of its composition. For instance, in the problem, a blend composed of 50% of crude A and 50% of crude B has a sulfur concentration of 0.035 which does not meet the specification for crude mix X nor for crude mix Y.

*2.5.1. Assignment Constraints.* In the SOS model, exactly one operation has to be assigned to each priority slot,

$$\sum_{v \in W} Z_{iv} = 1, \quad i \in T. \tag{6}$$

*2.5.2. Variable Constraints.* Variable constraints are given by their definitions. Start time, duration, and global volume variables are defined with big-$M$ constraints,

$$\begin{aligned} S_{iv} + D_{iv} &\leq H \cdot Z_{iv}, \quad i \in T,\ v \in W, \\ V_{iv}^{t} &\leq \overline{V_{v}^{t}} \cdot Z_{iv}, \quad i \in T,\ v \in W, \\ V_{iv}^{t} &\geq \underline{V_{v}^{t}} \cdot Z_{iv}, \quad i \in T,\ v \in W. \end{aligned} \tag{7}$$

Crude volume variables are positive variables whose sum equals the corresponding total volume variable,

$$\sum_{c \in C} V_{ivc} = V_{iv}^{t}. \tag{8}$$

Total and crude level variables are defined by adding to the initial level in the tank all inlet and outlet transfer volumes of operations of higher priority than the considered priority slot,

$$L_{ir}^{t} = L_{0r}^{t} + \sum_{j \in T, j<i} \sum_{v \in I_r} V_{iv}^{t} - \sum_{j \in T, j<i} \sum_{v \in O_r} V_{iv}^{t}, \quad i \in T,\ r \in R, \tag{9}$$

$$L_{irc} = L_{orc} + \sum_{j \in T, j<i} \sum_{v \in I_r} V_{ivc} - \sum_{j \in T, j<i} \sum_{v \in O_r} V_{ivc}, \quad i \in T,\ r \in R,\ c \in C. \tag{10}$$

*2.5.3. Sequencing Constraints.* Sequencing constraints restrict the set of possible sequences of operations. Cardinality and unloading sequence constraints are specific cases of sequencing constraints. More complex sequencing constraints will also be discussed later.

*2.5.4. Cardinality Constraint.* Each crude oil marine vessel has to unload its content exactly once. $\sum_{i \in T} \sum_{v \in O_r} Z_{iv} = 1, r \in R_V$. The total number of distillation operations is bounded by $\underline{N_D}$ and $\overline{N_D}$ in order to reduce the cost of CDU switches,

$$\underline{N_D} \leq \sum_{i \in T} \sum_{v \in W_D} Z_{iv} \leq \overline{N_D}. \tag{11}$$

*2.5.5. Unloading Sequence Constraint.* Marine vessels have to unload in order of arrival to the refinery. Considering two vessels $r_1, r_2 \in R_V, r_1 < r_2$ signifies that $r_1$ unloads before $r_2$,

$$\sum_{j \in T, j<i} \sum_{v \in O_{r_2}} Z_{jv} + \sum_{j \in T, j \geq i} \sum_{v \in O_{r_1}} Z_{jv} \leq 1. \tag{12}$$

*2.5.6. Scheduling Constraints.* Scheduling constraints restrict the values taken by time variables according to logistics rules.

*2.5.7. Nonoverlapping Constraint.* A nonoverlapping constraint between two sets of operations $W_1 \subset W$ and $W_2 \subset W$ states that any pair of operations $(v_1, v_2) \subset W_1 \times W_2$ must not be executed simultaneously.

Unloading operations must not overlap,

$$\sum_{v \in W_U} (S_{iv} + D_{iv}) \leq \sum_{v \in W_U} S_{jv} + H \cdot \left(1 - \sum_{v \in W_U} Z_{jv}\right), \quad i, j \in T,\ i < j. \tag{13}$$

Inlet and outlet transfer operations on a tank must not overlap,

$$\begin{gathered}\sum_{v\in I_r}\left(S_{iv}+D_{iv}\right)\leq\sum_{v\in O_r}S_{jv}+H\cdot\left(1-\sum_{v\in O_r}Z_{jv}\right),\\ i,j\in T,\ i<j,\ r\in R_S\cup R_C,\\ \sum_{v\in O_r}\left(S_{iv}+D_{iv}\right)\leq\sum_{v\in I_r}S_{jv}+H\cdot\left(1-\sum_{v\in I_r}Z_{jv}\right),\\ i,j\in T,\ i<j,\ r\in R_S\cup R_C.\end{gathered}\tag{14}$$

Although we do not consider crude settling in storage tanks after vessel unloading, it could be included in the model with a modified version of constraint (14) taking into account transition times. We define $\mathrm{TR}_V$ as the transition time after unloading operation $v \in W_U$ and TR as the maximum transition time, $\mathrm{TR} = \max_{v\in W_U}\mathrm{TR}_V$

$$\begin{gathered}\sum_{v\in I_r}\left(S_{iv}+D_{iv}+\mathrm{TR}_v\cdot Z_{iv}\right)\\ \leq\sum_{v\in O_r}S_{jv}+(H+\mathrm{TR})\cdot\left(1-\sum_{v\in O_r}Z_{jv}\right).\end{gathered}\tag{15}$$

Constraint (15) is valid in the four possible cases:

$$\begin{gathered}\left(\exists v_1\in I_r, Z_{iv_1}=1\right)\\ \wedge\left(\exists v_2\in O_r, Z_{jv_2}=1\right)\Longrightarrow S_{iv}+D_{iv_1}+\mathrm{TR}_{v_1}\leq S_{jv_2},\\ \left(\exists v_1\in I_r, Z_{iv_1}=1\right)\\ \wedge\left(\bigvee v_2\in O_r, Z_{jv_2}=1\right)\Longrightarrow S_{iv}+D_{iv_1}\leq H+\mathrm{TR}-\mathrm{TR}_{v_1},\\ \left(\bigvee v_1\in I_r, Z_{iv_1}=0\right)\\ \wedge\left(\exists v_2\in O_r, Z_{jv_2}=1\right)\Longrightarrow 0\leq S_{jv_2},\\ \left(\bigvee v_1\in I_r, Z_{iv_1}=0\right)\\ \wedge\left(\bigvee v_2\in O_r, Z_{jv_2}=0\right)\Longrightarrow 0\leq H+\mathrm{TR}.\end{gathered}\tag{16}$$

A tank may charge only one CDU at a time,

$$\begin{gathered}\sum_{v\in O_r}\left(S_{iv}+D_{iv}\right)\leq\sum_{v\in O_r}S_{jv}+H\cdot\left(1-\sum_{v\in O_r}Z_{jv}\right),\\ i,j\in T,\ i<j,\ r\in R_C.\end{gathered}\tag{17}$$

A CDU may be charged by only one tank at a time,

$$\begin{gathered}\sum_{v\in I_r}\left(S_{iv}+D_{iv}\right)\leq\sum_{v\in I_r}S_{jv}+H\cdot\left(1-\sum_{v\in I_r}Z_{jv}\right),\\ i,j\in T,\ i<j,\ r\in R_D.\end{gathered}\tag{18}$$

To avoid schedules in which a transfer is being performed twice at a time, thus possibly violating the flow rate limitations, constraint (19) is included in the model,

$$S_{iv}+D_{iv}\leq S_{jv}+H\cdot\left(1-Z_{jv}\right),\quad i,j\in T,\ i<j,\ v\in W.\tag{19}$$

*2.5.8. Continuous Distillation Constraint.* It is required that CDUs operate without interruption. As CDUs perform only one operation at a time, the continuous operation constraint is defined by equating the sum of the duration of distillations to the time horizon,

$$\sum_{i\in T}\sum_{v\in I_r}D_{iv}=H,\quad r\in R_D.\tag{20}$$

*2.5.9. Resource Availability Constraint.* Unloading of crude oil vessels may start only after arrival to the refinery. Let $S_r$ be the arrival time of vessel $r$,

$$S_{iv}\geq S_r\cdot Z_{iv},\quad i\in T,\ r\in R_v,\ v\in O_r.\tag{21}$$

*2.5.10. Operation Constraints.* Operation constraints restrict the values taken by operation and time variables according to operational rules.

*2.5.11. Flow Rate Constraint.* The flow rate of transfer operation $v$ is bounded by $\underline{\mathrm{FR}_v}$ and $\overline{\mathrm{FR}_v}$

$$\underline{\mathrm{FR}_v}\cdot D_{iv}\leq V_{iv}^t\leq\overline{\mathrm{FR}_v}\cdot D_{iv},\quad i\in T,\ v\in W.\tag{22}$$

*2.5.12. Property Constraint.* The property $k$ of the blended products transferred during operation $v$ is bounded by $\underline{x_{vk}}$ and $\overline{x_{vk}}$. The property $k$ of the blend is calculated from the property $x_{ck}$ of crude $c$ assuming that the mixing rule is linear,

$$\underline{x_{vk}}\cdot V_{iv}^t\leq\sum_{c\in C}x_{ck}V_{ivc}\leq\overline{x_{vk}}\cdot V_{iv}^t,\quad i\in T,\ v\in W,\ k\in K.\tag{23}$$

*2.5.13. Composition Constraint.* It has been shown that processes including both mixing and splitting of streams cannot be expressed as a linear model. Mixing occurs when two streams are used to fill a tank and is expressed linearly in constraint (10). Splitting occurs when partially discharging a tank, resulting in two parts: the remaining content of the tank and the transferred products. This constraint is nonlinear. The composition of the products transferred during a transfer operation must be identical to the composition of the origin tank,

$$\frac{L_{irc}}{L_{ir}^t}=\frac{V_{ivc}}{V_{iv}^t},\quad i\in T,\ r\in R,\ v\in O_r,\ c\in C.\tag{24}$$

Constraint (24) is reformulated as an equation involving bilinear terms,

$$V_{ivc}\cdot L_{ir}^t=L_{irc}\cdot V_{iv}^t,\quad i\in T,\ r\in R,\ v\in O_r,\ c\in C.\tag{25}$$

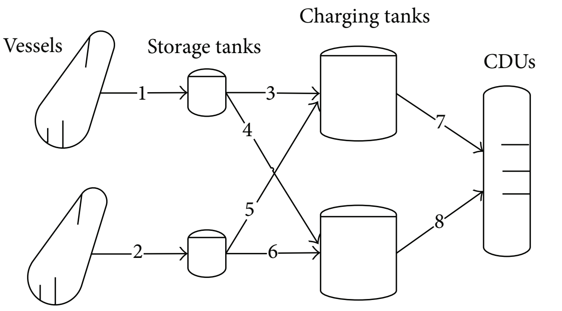


Figure 1: Crude oil operations system for the problem.

Note that constraint (25) is correct even when operation $v$ is not assigned to priority-slot $i$, as then

$$V_{iv}^t = V_{ivc} = 0. \tag{26}$$

*2.5.14. Resource Constraints.* Resource constraints restrict the use of resources throughout the scheduling horizon.

*2.5.15. Tank Capacity Constraint.* The level of materials in the tank $r$ must remain between minimum and maximum capacity limits $\underline{L_r^t}$ and $\overline{L_r^t}$, respectively. Let $L_{0r}^t$ be the initial total level and let $L_{0rc}$ be the initial level of crude $c$ in the tank $r$. As simultaneous charging and discharging of tanks is forbidden, the following constraints are sufficient:

$$\begin{gathered}
\underline{L_r^t} \leq L_{ir}^t \leq \overline{L_r^t}, \quad i \in T,\ r \in R_S \cup R_C, \\
0 \leq L_{irc} \leq \overline{L_r^t}, \quad i \in T,\ r \in R_S \cup R_C,\ c \in C, \\
\underline{L_r^t} \leq L_{0r}^t + \sum_{i \in T} \sum_{v \in I_r} V_{iv}^t - \sum_{i \in T} \sum_{v \in O_r} V_{iv}^t \leq \overline{L_r^t}, \\
r \in R_S \cup R_C, \\
0 \leq L_{0rc} + \sum_{i \in T} \sum_{v \in I_r} V_{ivc} - \sum_{i \in T} \sum_{v \in O_r} V_{ivc} \leq \overline{L_r^t}, \\
r \in R_S \cup R_C,\ c \in C.
\end{gathered} \tag{27}$$

*2.5.16. Demand Constraint.* Demand constraints define lower and upper limits, $\underline{D_r}$ and $\overline{D_r}$, on total volume of products transferred out of each charging tank $r$ during the scheduling horizon,

$$\underline{D_r} \leq \sum_{i \in T} \sum_{v \in O_r} V_{iv}^t \leq \overline{D_r}, \quad r \in R_C. \tag{28}$$

## 3. Finite State Theory

This section presents in a somewhat informal way those basic notions and definitions from formal language and finite state theories, which are relevant for the sections to follow. Related definitions are taken from literature [20, 21]. Readers, who are unfamiliar with formal language theory, are advised to consult the sources whenever necessary.

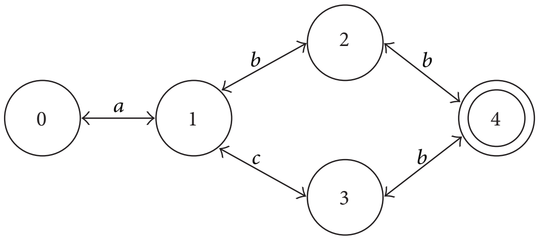


Figure 2: A deterministic finite state automaton (DFA).

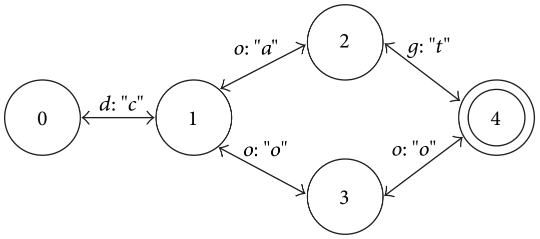


Figure 3: Finite state transducer encoding the relation {(dog, cat), (dog, cow)}.

*3.1. Finite State Automata.* A DFA is a 5-tuple $(Q, \Sigma, \delta, i, F)$, where $Q$ is a set of states, $\Sigma$ is an alphabet, $i$ is the initial state, $F \subseteq Q$ is a set of final states, and $\delta$ is a transition function mapping $Q \times \Sigma$ to $Q$. That is, for each state $u$ and symbol $a$, there is at most one state that can be reached from $u$ by "following" $a$ (Figure 2).

*3.2. Finite State Transducers.* A finite state transducer (FST) is a 6-tuple $(\Sigma_1, \Sigma_2, Q, \delta, i, F)$, where $Q$, $i$, and $F$ are the same as for DFA, $\Sigma_1$ is input alphabet, $\Sigma_2$ is output alphabet, and $\delta$ is a function mapping $Q \times (\Sigma_1 \cup \{\varepsilon\}) \times (\Sigma_2 \cup \{\varepsilon\})$ to a subset of the power set of $Q$ (Figure 3). Intuitively, an FST is much like an NFA except that transitions are made on strings instead of symbols and, in addition, they have outputs.

*3.3. Finite State Calculus.* As argued in Karttunen [22–25], many of the rules used can be analyzed as special cases of regular expressions. They extend the basic regular expression with new operators. These extensions make the finite state automation and finite state transducer become more suitable for particular applications. The system described below was implemented using FSA Utilities [26], a package for implementing and manipulating finite state automata, which provides possibilities for defining new regular expression operators. The part of FSAs built in regular expression syntax relevant to this paper is listed in Table 4.

One particular useful extension of the basic syntax of regular expressions is the replace-operator. Karttunen [22–25] argues that many phonological and morphological rules can be interpreted as rules which replace a certain portion of the input string. Although several implementations of the replace-operator are proposed, the most relevant case for our purposes is the so-called "leftmost longest-match" replacement. In case of overlapping rule targets in the input, this operator will replace the leftmost target, and in cases where

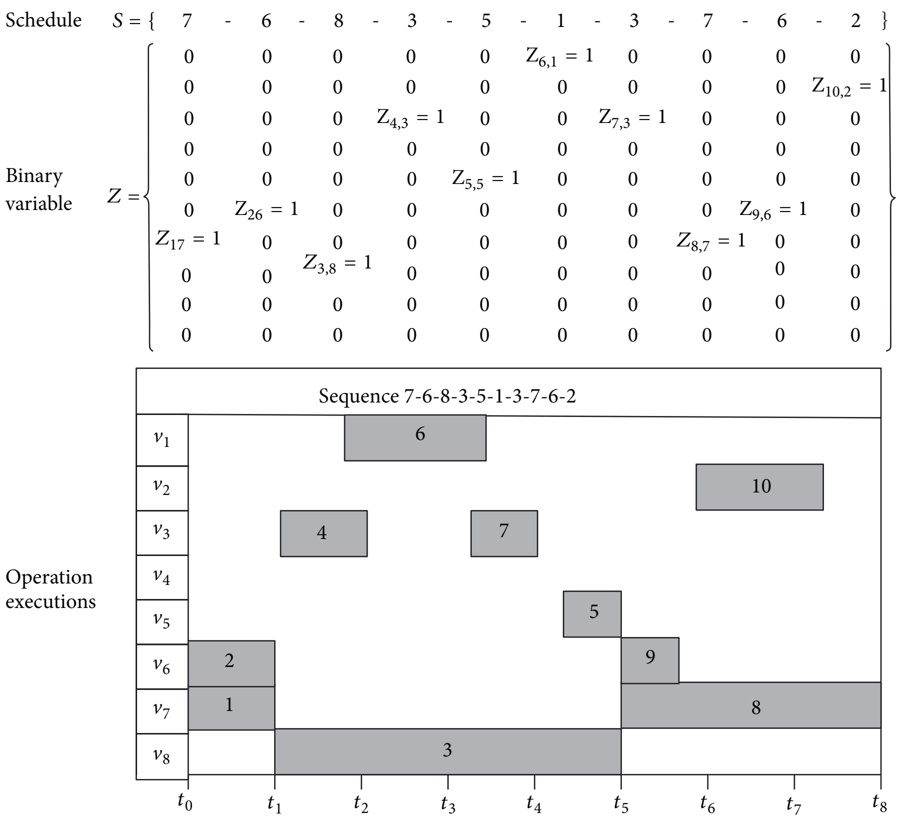


Figure 4: An example to indicate the relationship between binary variable and schedule.

a rule target contains a prefix which is also a potential target, the longer sequence will be replaced. Gerdemann and van Noord [27] implement leftmost longest-match replacement in FSA as the operator:

$$\text{replace}\ (\text{Target}, \text{LeftContext}, \text{RightContext}), \qquad (29)$$

where Target is a transducer defining the actual replacement and LeftContext and RightContext are regular expressions defining the left and right context of the rule, respectively. The segmentation task discussed in the mutation procedure makes crucial use of longest-match replacement.

## 4. The Hybrid Algorithm

From the point view of optimization efficiency and robustness, a novel two-level optimization framework based on finite state method and GA is proposed for the MINLP model in this section.

*4.1. Two-Level Optimization Structure.* As the foundation of the framework, a two-level optimization structure is introduced. Once all binary variables are fixed the original problem becomes a relatively simpler model with only continuous variable. Following this deal, we rewrite (5) as follows:

$$\max\left(J\left(\xi, z\right)\right) \Longleftrightarrow \max_{z}\left[\max_{\xi} J\left(\xi, \overline{z}\right)\right], \qquad (30)$$

where $\xi$ and $z$ represent continuous and binary variables, respectively. Equation (30) shows when $z$ is fixed as $\overline{z}$, the submodel $J(\xi, \overline{z})$ can be solved optimally by continuous-optimization solvers in the inner level; then we update $\overline{z}$ towards the best binary solution $z^*$ in the outer level.

We used an example in Figure 4 to show how binary solution can be mapped to a scheduling sequence. The schedule $S = [7683513762]$ where 7 stands for the specific operation 7 to assign to position 1 corresponding to the binary decisions $Z_{17} = 1$.

*4.2. Initial Population.* Based on the sequencing rules [18] and the extension to the regular expression calculus [22–25], a DFA model which builds legal schedules complying with sequencing rules and operation condition is constructed. The whole set of possible schedules is too huge to be processed at once. The DFA model of the schedule constitutes a reasonable framework, capturing all possible schedules and removing many redundant sequences of operations. Initial values of decision variables must satisfy the equality constraints and operation condition and therefore represent a feasible operating point.

Here, we still use the instance with 8 operations from Mouret et al. [18] to describe an efficient sequencing rule by

```
macro (procedure,
        segmentation, % segmentation of the input sequence into a set of sub-sequence
        mutation, % apply mutation rules
        clean up) % remove markers
```

Algorithm 1

using a regular expression. A feasible sequence $v_1 \cdots v_i \cdots v_n$ can be described by the following:

$$\begin{gathered} \text{sequence} = (\varepsilon + L_a)(L_b \cdot L_a)^* (\varepsilon + L_b), \\ L_a = 7(\varepsilon + 4)(\varepsilon + 6)(\varepsilon + 1 + 14)(\varepsilon + 2 + 26), \\ L_b = 8(\varepsilon + 3)(\varepsilon + 5)(\varepsilon + 1 + 13)(\varepsilon + 2 + 25). \end{gathered} \tag{31}$$

However, this automation suffers from a serious problem of overgeneration. For example, the short length of the sequence may lead to infeasibility, while the long length of the sequence may result in an unsolvable model. It is an interesting challenge for finite state syntactic description to specify a sublanguage that contains all and only the sequences of valid length.

Our solution is to construct a suitable constraint for the sequences of valid length. The constraint expressions denote a language that admits sequences of valid length but excludes all others. We obtain the desired effect by intersecting the constraint language with the original language of sequence expressions. The intersection of the two languages contains all and only the valid dates:

$$\text{ValidSequence} = \text{Sequence} \cap \text{ValidLength}. \tag{32}$$

The ValidLength constraint is a language that includes all sequences of length $n$:

$$\text{ValidLength} = (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8)^n. \tag{33}$$

We have now completed the task of describing the language of valid sequences from the set of possible sequence expressions. It is also possible to create an automation on the basis of the regular expression and ValidSequence and then generate all possible sequences $v_1 \cdots v_i \cdots v_n$ accepted by the automaton. The processes are implemented using FSA Utilities [26] that is a package for implementing and manipulating DFA and finite state transducer. In order to generate all possible sequences. When all possible sequences $v_1 \cdots v_i \cdots v_n$ accepted by the automaton are generated, and the population of the according possible binary decisions is generated. In the initial population stage of GA, the population size is the number of individuals. When the number of individuals is given, a population of candidate solutions is generated by randomly selecting from the population of the all possible binary decisions.

*4.3. Rule-Based Mutation Approach.* In the mutation stage, we use a finite state transducer for this rule-based mutation process. The rule-based mutation strategy must obey the sequencing rule and the nonoverlapping constraint such that all involved solutions in GA are feasible.

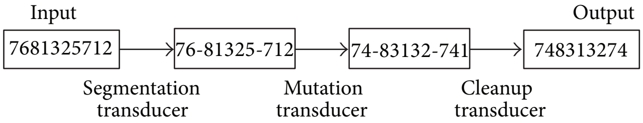


Figure 5: An example of mutation.

```
replace (
        [identity(SSequence),[]x-],[],]
        ).
```

Algorithm 2

The proposed mutation approach is a two-step procedure.

*Step 1.* Segmentation of the input sequence into a set of subsequences (i.e., the subsequence which belongs to the regular language L7 or L8).

*Step 2.* Mutation of the subsequences into others.

Formally, the rule-based mutation procedure is implemented as the composition of three transducers (see Algorithm 1).

An example of mutation including the intermediate steps is given for the sequence "7681325712" as shown in Figure 5.

*4.3.1. Segmentation Transducer.* Segmentation transducer splits an input sequence into subsequences. The goal of segmentation is to provide a convenient representation level for the next mutation step.

Segmentation is defined as shown in Algorithm 2.

The macro "SSequence" defines the set of subsequences. The subsequences which belong to the regular language L7 and L8 are displayed in Tables 1 and 2. Segmentation attaches the marker "–" to each subsequence. The Targets are identified using leftmost longest-match, and thus at each point in the input, only the longest valid segment is marked.

*4.3.2. The Mutation Rules.* In the GA process, the mutation rules are made by carefully considering nonoverlapping constraint between operations. A concrete instance for partially illustrating the mutation rules is given in Algorithm 3. Note that the final element of the left-context must be a marker and

```
marco (conversion, $L_7$_subsequence_rules)
                        $\circ$ $L_8$_subsequence_rules)
marco ($L_7$_subsequence_rules,
replace ($\{2, 4, 6\} \times 1, 7,$ —)
$\circ$ replace ($\{14, 26, 41, 42, 46, 61, 62\} \times 12, 7,$ —)
$\circ$ replace ($\{142, 412, 414, 426, 461, 462, 612, 614, 626\} \times 126, 7,$ —)
marco ($L_8$_subsequence_rules,
replace ($\{2, 3, 5\} \times 1, 8,$ —)
$\circ$ replace ($\{13, 25, 31, 32, 35, 51, 52\} \times 12, 8,$—)
$\circ$ replace ($\{132, 312, 313, 325, 351, 352, 512, 513, 525\} \times 125, 8,$ —)
```

Algorithm 3: An example to demonstrate the mutation rules.

Table 1: Subsequence belonging to $L_7$.

| Length | Sequences belonging to $L_7$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 7 | | | | | | | | | |
| 2 | 71 | 72 | 74 | 76 | | | | | | |
| 3 | 712 | 714 | 726 | 741 | 742 | 746 | 761 | 762 | | |
| 4 | 7126 | 7142 | 7412 | 7414 | 7426 | 7461 | 7462 | 7612 | 7614 | 7626 |
| 5 | 71426 | 74126 | 74142 | 74612 | 74614 | 74626 | 76126 | 76142 | | |
| 6 | 741426 | 746126 | 746142 | 761426 | | | | | | |
| 7 | 7461426 | | | | | | | | | |

the target itself ends in "–." This ensures that mutation rules cannot apply to the same subsequence.

## 5. Experimental Study

In this section, the same problem from the literature [18] is used for computational experiments. The proposed methodology is compared with existing promising algorithms, mixed-coding GA [15, 28]. Figure 1 depicts the refinery configuration for problem. The data involved in the problem are given in Table 3. The performance comparison with different computing times, such as 350 s, 500 s,. . ., 2400 s, is conducted. The objective value is used to statistically analyze the optimization results.

The performance comparison between the two methodologies used is illustrated in Figure 6, which shows that the hybrid optimization algorithm which combined the finite state method and GA will statistically outperform the mixed-coding counterpart. The genetic algorithm which combined the finite state method and GA finds feasible solutions very fast and is able to find better solutions in reasonable time.

In Figure 7, we compare the objective variance of each iteration in the two evolution processes of these two kinds of methodology. By tracking the evolution process, we find that the mixed-coding GA is easy to stick in a local minimal sequence solution. This situation only can be improved through increasing the mutation scaling factor. However, this may result in a hard convergence, unless sufficient iterations are implemented. As for the hybrid optimization algorithm, the optimization processes of binary variable and continuous variable are separated. The performance of the whole methodology mainly depends on the FSM which captures most promising schedules and removes many redundant sequences of operations, so that the user can use a small population size of corresponding discrete variables to obtain suboptimal solutions. From Figure 7, we see that the proposed method has converged at 350 iterations as opposed to 2400 iterations for the mixed-coding GA.

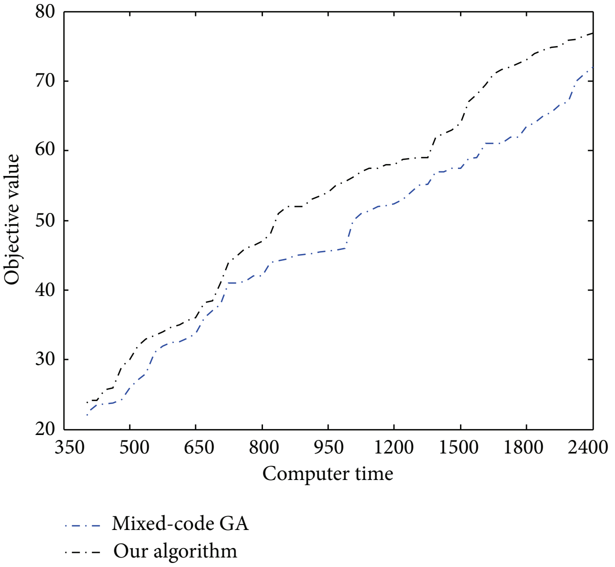


Figure 6: Average objective values of two methodologies.

The success of the proposed algorithm lies in a comprehensive analysis of the region of the search space and its capacity to focus the search on the regions with the partial solution. One of the good merits of the hybrid algorithm is

Table 2: Subsequence belonging to $L_8$.

| Length | Sequences belonging to $L_8$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 8 | | | | | | | | | |
| 2 | 81 | 82 | 83 | 85 | | | | | | |
| 3 | 812 | 813 | 825 | 831 | 832 | 835 | 851 | 852 | | |
| 4 | 8125 | 8132 | 8312 | 8313 | 8325 | 8351 | 8352 | 8512 | 8513 | 8525 |
| 5 | 81325 | 83125 | 83132 | 83512 | 83513 | 83525 | 85132 | 85125 | | |
| 6 | 831325 | 835125 | 835132 | 851325 | | | | | | |
| 7 | 8351325 | | | | | | | | | |

Table 3: Problem data.

| Scheduling horizon | | | 8 days | | |
|---|---|---|---|---|---|
| Vessels | Arrival time | | Composition | Amount of crude | |
| Vessel 1 | 0 | | 100% A | 1000 | |
| Vessel 2 | 4 | | 100% B | 1000 | |
| Storage tanks | Capacity | | Initial composition | Initial amount | |
| Tank 1 | [0, 1000] | | 100% A | 250 | |
| Tank 2 | [0, 1000] | | 100% B | 750 | |
| Charging tanks | Capacity | | Initial composition | Initial amount | |
| Tank 1 (mix X) | [0, 1000] | | 100% C | 500 | |
| Tank 1 (mix X) | [0, 1000] | | 100% D | 500 | |
| Crudes | 1 | Gross margin | Crude mixtures | Property1 | Demand |
| Crude A | 0.01 | 9 | Crude mix X | [0.015, 0.025] | [1000, 1000] |
| Crude B | 0.06 | 4 | Crude mix Y | [0.045, 0.055] | [1000, 1000] |
| Crude C | 0.02 | 8 | Unloading flow rate | [0, 500] | |
| Crude D | 0.05 | 5 | transfer flow rate | [0, 500] | |

Table 4: A fragment of FSA regular expression syntax and $U$ transducers, and $R$ can be either.

| | |
|---|---|
| []: | The empty string |
| $[R_1, \ldots, R_n]$: | Concatenation |
| $\{R_1, \ldots, R_n\}$ | Disjunction |
| $R^{\wedge}$: | Optionality |
| Identity ($A$): | Identity: the transducer which maps each element in $A$ onto itself |
| $T \circ U$: | Composition of the transducers $T$ and $U$ |
| macro (Term, $R$): | Use term as an abbreviation for $R$ |

that each solution involved in the GA algorithm is guaranteed to be feasible by using the mutation rules generated by DFM method while in existing GA algorithms the procedure to generate feasible solution under complex process constraints is very time costive. The deterministic finite automata (DFA) can easily represent this kind of structure. Furthermore, the complex process constraints can be very difficult to express with mixed integer programming. Consequently, it is unfeasible to solve the industrial problem by using MIP solver.

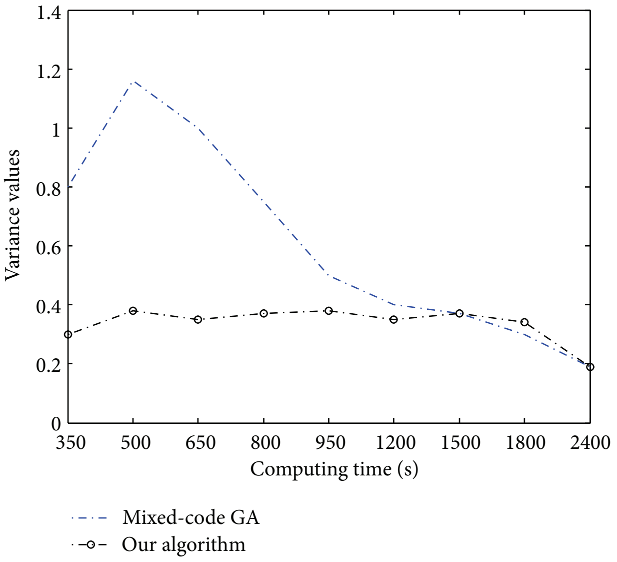


Figure 7: Variance values of two methodologies.

## 6. Conclusion

In this paper, a novel hybrid optimization algorithm which combined the finite state method and GA is proposed. The proposed algorithm constitutes a reasonable framework, capturing both the operating condition and sequencing rule of the schedule. The solution captures all possible schedules and removes many redundant sequences of operations. The algorithm is equivalent to introducing new structure information into the optimization process, which will help reduce the risk of trapping in a local minimal sequence solution. The hybrid optimization algorithm is an effective and robust tool to solve the crude oil scheduling problem in terms of efficiency and reliability. Algorithms only with the two properties are suitable for solving practical engineering application.

## 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 the China National Natural Science Foundation under Grant 61203178, Grant 61304214, and Grant 61290323. The authors thank the financial funds from Shanghai Science and Technology Committee under Grant 12511501002 and Grant 13511501302.

## References

[1] J. M. Pinto, M. Joly, and L. F. L. Moro, “Planning and scheduling models for refinery operations,” *Computers and Chemical Engineering*, vol. 24, no. 9-10, pp. 2259–2276, 2000.

[2] J. Li, I. A. Karimi, and R. Srinivasan, “Recipe determination and scheduling of gasoline blending operations,” *AIChE Journal*, vol. 56, no. 2, pp. 441–465, 2010.

[3] J. Li, R. Misener, and C. A. Floudas, “Continuous-time modeling and global optimization approach for scheduling of crude oil operations,” *AIChE Journal*, vol. 58, no. 1, pp. 205–226, 2012.

[4] Z. Jia, M. Ierapetritou, and J. D. Kelly, “Refinery short-term scheduling using continuous time formulation: crude-oil operations,” *Industrial and Engineering Chemistry Research*, vol. 42, no. 13, pp. 3085–3097, 2003.

[5] C. A. Méndez, I. E. Grossmann, I. Harjunkoski, and P. Kaboré, “A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations,” *Computers and Chemical Engineering*, vol. 30, no. 4, pp. 614–634, 2006.

[6] M. Pan, X. Li, and Y. Qian, “New approach for scheduling crude oil operations,” *Chemical Engineering Science*, vol. 64, no. 5, pp. 965–983, 2009.

[7] M. F. Cardoso, R. L. Salcedo, S. F. de Azevedo, and D. Barbosa, “A simulated annealing approach to the solution of minlp problems,” *Computers and Chemical Engineering*, vol. 21, no. 12, pp. 1349–1364, 1997.

[8] I. E. Grossmann, “Review of nonlinear mixed-integer and disjunctive programming techniques,” *Optimization and Engineering*, vol. 3, no. 3, pp. 227–252, 2002.

[9] E. L. Lawler and D. E. Wood, “Branch-and-bound methods: a survey,” *Operations Research*, vol. 14, no. 4, pp. 699–719, 1966.

[10] M. A. Duran and I. E. Grossmann, “An outer-approximation algorithm for a class of mixed-integer nonlinear programs,” *Mathematical Programming*, vol. 36, no. 3, pp. 307–339, 1986.

[11] A. M. Geoffrion, “Generalized Benders decomposition,” *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 1972.

[12] C. D’Ambrosio and A. Lodi, “Mixed integer nonlinear programming tools: a practical overview,” *4OR*, vol. 9, no. 4, pp. 329–349, 2011.

[13] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[14] T. Yokota, M. Gen, and Y.-X. Li, “Genetic algorithm for non-linear mixed integer programming problems and its applications,” *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 905–917, 1996.

[15] L. Costa and P. Oliveira, “Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems,” *Computers and Chemical Engineering*, vol. 25, no. 2-3, pp. 257–266, 2001.

[16] J. M. Ponce-Ortega, M. Serna-González, and A. Jiménez-Gutiérrez, “Heat exchanger network synthesis including detailed heat exchanger design using genetic algorithms,” *Industrial and Engineering Chemistry Research*, vol. 46, no. 25, pp. 8767–8780, 2007.

[17] K.-M. Björk and R. Nordman, “Solving large-scale retrofit heat exchanger network synthesis problems with mathematical optimization methods,” *Chemical Engineering and Processing: Process Intensification*, vol. 44, no. 8, pp. 869–876, 2005.

[18] S. Mouret, I. E. Grossmann, and P. Pestiaux, “A novel priority-slot based continuous-time formulation for crude-oil scheduling problems,” *Industrial and Engineering Chemistry Research*, vol. 48, no. 18, pp. 8515–8528, 2009.

[19] H. Lee, J. M. Pinto, I. E. Grossmann, and S. Park, “Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management,” *Industrial and Engineering Chemistry Research*, vol. 35, no. 5, pp. 1630–1641, 1996.

[20] J. E. Hopcroft, *Introduction to Automata Theory, Languages, and Computation*, Pearson Education, India, New Delhi, India, 3rd edition, 2008.

[21] E. Roche and Y. Schabes, *Finite-State Language Processing*, The MIT Press, Cambridge, Mass, USA, 1997.

[22] L. Karttunen, “Constructing lexical transducers,” in *Proceedings of the 15th conference on Computational Linguistics*, vol. 1, Association for Computational Linguistics, 1994.

[23] L. Karttunen, “The replace operator,” in *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 1995.

[24] L. Karttunen, “Directed replacement,” in *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 1996.

[25] L. Karttunen and K. R. Beesley, *Two-Level Rule Compiler*, Xerox Corporation, Palo Alto Research Center, 1992.

[26] G. van Noord, “FSA utilities: a toolbox to manipulate finite-state automata,” in *Automata Implementation*, pp. 87–108, Springer, New York, NY, USA, 1997.

[27] D. Gerdemann and G. van Noord, “Transducers from rewrite rules with backreferences,” in *Proceedings of the 9th Conference on European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 1999.

[28] Y.-C. Lin, K.-S. Hwang, and F.-S. Wang, “A mixed-coding scheme of evolutionary algorithms to solve mixed-integer nonlinear programming problems,” *Computers and Mathematics with Applications*, vol. 47, no. 8-9, pp. 1295–1307, 2004.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 286575, 15 pages
http://dx.doi.org/10.1155/2014/286575

*Research Article*

# Genetic Algorithm Application in Optimization of Wireless Sensor Networks

**Ali Norouzi and A. Halim Zaim**

*Computer Engineering Department, Istanbul University, 34320 Istanbul, Turkey*

Correspondence should be addressed to Ali Norouzi; norouzi@cscrs.itu.edu.tr

Received 31 August 2013; Accepted 19 November 2013; Published 16 February 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

There are several applications known for wireless sensor networks (WSN), and such variety demands improvement of the currently available protocols and the specific parameters. Some notable parameters are lifetime of network and energy consumption for routing which play key role in every application. Genetic algorithm is one of the nonlinear optimization methods and relatively better option thanks to its efficiency for large scale applications and that the final formula can be modified by operators. The present survey tries to exert a comprehensive improvement in all operational stages of a WSN including node placement, network coverage, clustering, and data aggregation and achieve an ideal set of parameters of routing and application based WSN. Using genetic algorithm and based on the results of simulations in NS, a specific fitness function was achieved, optimized, and customized for all the operational stages of WSNs.

## 1. Introduction

WSNs are constituted of small sensors with specialized applications and limitations designed for specific purposes. The applications are divided into military, commercial, and medical applications. Among military applications are communication, command, and intelligence defense networks. Health care system for disables in remote areas, smart environment for the elderly, physicians, and medical staff communication networks, and patient surveillance systems are some of medical applications. Moreover, there is a wide range of commercial applications including security systems, fire safety systems, environment pollution monitor systems (chemical, microbial, and nuclear pollutions), vehicle tracking, supervising and controlling systems, traffic control system, and natural disasters studies (e.g., earthquake and flood) [1]. Wide range of applications has resulted in development of variety of protocols which include plenty of flexible parameters. At any rate, some parameters, due to their wide range of utilization, can be found in several applications (as common parameters) and of great importance. Wireless sensor networks use mobile energy sources and rechargeable batteries, and due to technological limitations, these batteries can supply energy for a short period of time. Thus, optimum utilization of energy in such networks is of great importance [2].

Necessity of data integrity in WNSs due to support continuous and permanent communication among the sensors has made the lifetime another important parameter in WSNs. The present study surveys some specific parameters throughout different operational stages of WSNs. In general, operational stages of classic WSNs are divided into node placement, network coverage, clustering, and data aggregation. Figure 1(a) pictures general classification of the main operational stages of WSNs.

An important stage before establishment of a WSN is "node placement." Generally, there are several types of node distributions in WSNs including regular, random, and grid distributions. Under grid layout, the distance between each node can be estimated. An example of grid layout is pictured in Figure 1(b) and, clearly, the gap between the nodes is fixed.

The decision about type of layout depends on the expected application, so that nodes for military purposes are usually scattered by airplanes over military zones, while in case of underwater sensors, regular distribution is adopted and grid layouts are usually used for urban networks (Figure 1(b)).

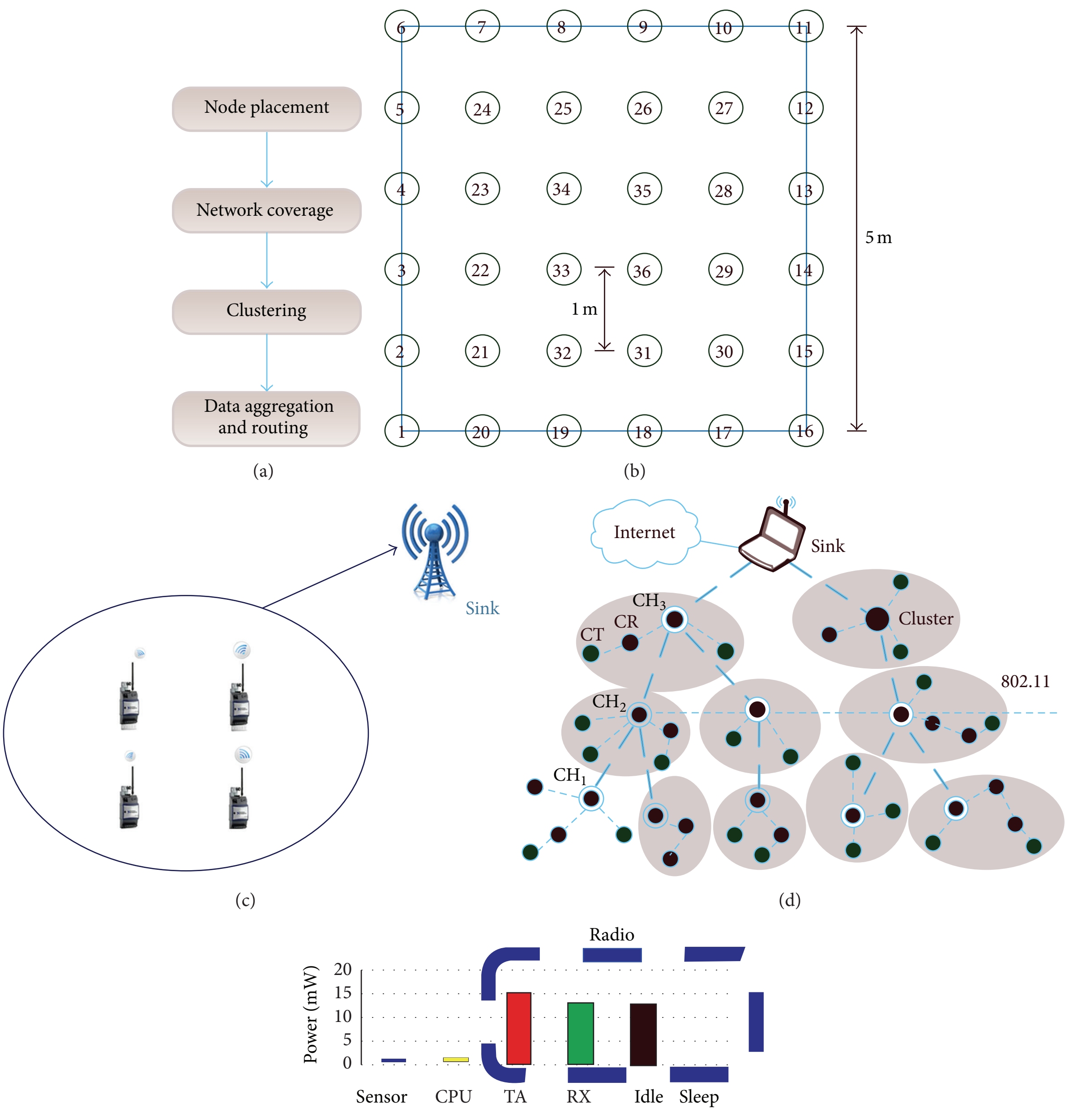


Figure 1: (a) Main operatioanl stages of WSNs. (b) Grid layout of WSN. (c) Network coverage stage. (d) Clustering in WSNs. (e) Energy consumption in different states of WSNs.

Poisson's distribution is useful for modeling different types of random phenomena; it generates an estimate of binomial probabilities. In addition to an estimator distribution, Poisson's distribution is a useful probability model for the events that happen randomly whether in time or place. The distribution is usually used for detailed study on and simulation of wireless networks.

The next stage is to connect the sensors based on the range of service. As mentioned before, among different features of sensors, radio range and service domain are key factors. Taking into account the required area coverage, the best layout must be adopted to reach the best quantity and quality of the services. Ineffective layout means waste of energy and financial resources.

Clustering is another main operation, which plays a key role in WSN optimization. By clustering, the sensor nodes are divided into groups known as division cluster. Each cluster has a cluster head that aggregates data from the nodes in the cluster and forwards the data to the sink directly or step by step using other clusters' heads. Therefore, the nodes may reduce their communication heading compared with the situation when data are forwarded directly to the sink. That is, clustering is an effective approach to attenuate load between sensor nodes.

Figure 1(d) illustrates nodes clustering in WSNs. Clustering is a way to save more energy and increase lifetime of the sensors in WSNs. The technique also has other advantages such as improved security, less extra data, and improved scalability of the network. To achieve better performance, different protocols can be used depending on the application.

Babamir and Norouzi proposed an efficient aggregate signcryption scheme to maximize the security of data in a kind of wireless medical network named the disconnected or unattended wireless sensor network [3]. Also in other work, they proposed another new secure scheme in which various security goals such as confidentiality, authentication and integrity. In addition, the aggregation process of their scheme reduces the space and communication overheads both for sensors and sink. The proposed technique efficiently enables the sensors and sinks to protect, verify, and recover all the related data [4].

The protocols are reliable ways to increase lifetime of the networks, although they cause more energy consumption by the cluster heads. Therefore, to increase lifetime of the network, cluster heads must be reelected during each period of cluster layout. In spite of the fact that the protocols ensure implementation of an effective clustering algorithm, they fail to guarantee adoption of the best node as cluster head.

Through optimization, the algorithms may attenuate energy consumption to a great extent and consequently improve efficiency and lifetime lifetime of the network.

Eventually, the transfer of data and queries between the main stations, information sinks, or events is another important issue in WSNs. A simple process for transfer of data is the direct transfer of data between the node and base station. The single-step oriented process is too costly; the more the distance between the node and base station, the more energy is needed and consequently the shorter the lifetime of the network.

Another process for transfer of data is multistep oriented transfer for a specific radius. This process saves considerable deal of energy and lessens collision in the network to a great deal, although, depending on the place of using routing mechanisms, they have some limitations.

Main reason that makes researchers more interested in the issue data gathering and routing stages is the considerable energy consumed at this stage. Figure 1(e) pictures required energy in every states of the WSN. Clearly, the highest energy consumption is by radio communication. Therefore, more detailed studies on this stage hold great promises to optimize WSN concerning energy consumption and lifetime of the network [2, 5, 6].

Improvement of the parameters mentioned above eventuates in an optimized WSN. There are variety of methods to this end, such as fuzzy theory, neural networks, and evolutionary algorithm and thanks to its better results for larger scale networks and the fact that it generates final formula at the end, genetic algorithm is more common. The availability of final formula makes the algorithm more useful and helpful for human users. Thus, the present study uses genetic algorithm for optimization and customization of the networks [7].

This paper is organized as follows; Section 2 gives a brief description of genetic algorithm. Sections 3, 4, 5 and 6 present our proposed fitness function in node placement, network coverage, clustering, data aggregation, and details of algorithms, respectively. Finally, Section 7 presents conclusions and suggestions for future projects.

## 2. Genetic Algorithm

Also known as a global heuristic algorithm, a generic algorithm estimates an optimal solution through generating different individuals [8]. Focused fitness function is one of procedures of the algorithm. Following section describes the fundamental parts of a generic algorithm. Figure 2(a) indicates the general scheme of genetic algorithm mechanism.

*2.1. Initialization.* The genetic algorithm starts with an elementary population comprised of random chromosomes which includes genes with a sequence of 0 s or 1 s. Afterward, the algorithm leads individuals to achieve an optimum solution by the way of repetitive processes including crossover and selection operators. There are two ways to develop a new population [9]: steady-state GA and generational GA. In the case of the former, one or two members in the population are replaced and at the same time, the generational GA replaces all the generated individuals of a generation.

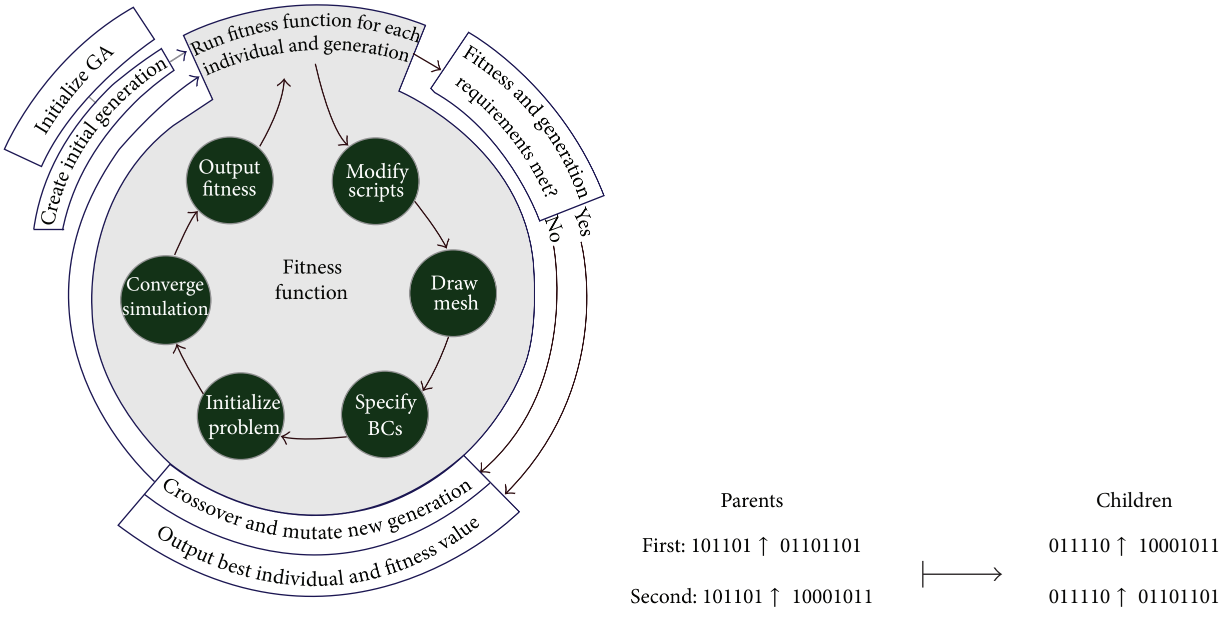
*2.2. Fitness.* Under the genetic algorithm, the fitness function, by definition, is a process for scoring each chromosome based on their qualification. The assigned score is a trait for continuation of further reproduction. Dependence to problem by the fitness function is considerable, so that in case of some problems, it is not possible to define the problem. Naturally, individuals are permitted to go to the new generation based on their fitness score. Therefore, the score dictates the fate of individuals.

*2.3. Selection.* During every successive generation, a new generation is developed through adopting members of the current generation to mate on the bases of their fitness. The individuals with higher fitness score have higher chance for being selected, the process which results in preferential adoption of the best solution. Majority of the functions include a stochastically designed element for adopting small number of less fit individuals for sake of keeping diversity in the population [10]. Among the many selection methods, Roulette-Wheel is adopted to differentiate proper individuals with the probability of

$$P_i = \frac{F_i}{\sum_{j=1}^{n} F_i}, \tag{1}$$

where $F_i$ and "$n$" are the fitness chromosome and the size of population, respectively. According to the Roulette-Wheel, each individual is assigned a value between 0 and 1.

*2.4. Crossover.* The crossover or reproduction process constitutes the major step toward production. Indeed, sexual


Initialize GA
Create initial generation
Run fitness function for each individual and generation
Fitness and generation requirements met?
No
Yes
Output fitness
Modify scripts
Fitness function
Converge simulation
Draw mesh
Initialize problem
Specify BCs
Crossover and mutate new generation
Output best individual and fitness value
Parents
Children
First: 101101 ↑ 01101101
Second: 101101 ↑ 10001011
011110 ↑ 10001011
011110 ↑ 01101101

(a)

(b)

Figure 2: (a) The general scheme of GA mechanism. (b) Single point method at random point 6.

reproductive process by wich inherited characteristics are transferred from one generation to the next generation,is simulated. In the reproduction process, crossover process adopts a couple of individuals as the parents through breeding selection process. The process continues to reach the desired size in the new population. Generally, several crossover operations take place, each of which with different aims. The easiest way is single point, where a random point is adopted to divide the role of the patents. One example of mating by two chromosomes in single point way is pictured in Figure 2(a).

Figure 2(b) represents two children that are from a single set of parents. The bit sequence of the offspring duplicates one parent's bit sequence until the crossover point. Afterward, the bit sequence of the other parent is replicated as the second part of children.

*2.5. Fitness Parameters in Wireless Sensor Network.* The fitness of a chromosome determines the extent to which the consumption of energy is minimized and coverage is maximized. In what follows, some important fitness parameters in WSN are discussed.

(1) Direct distance to base station (DDBS): it refers to the sum of direct distance between all sensornodes and the BS represented by $d_i$ as

$$\mathrm{DDBS} = \sum_{i=1}^{m} d_i, \tag{2}$$

where "$m$" stands for the number of nodes. Clearly, consumption of energy, reasonably, is subject to the number of nodes and for large WSN the energy is extreme. Moreover, DDBS is acceptable for smaller networks where number of close nodes is not considerable.

(2) Cluster based distance (CD): The total CHs and BS distances and the sumof the distance between the determined member nodes and their cluster heads (3).

$$\mathrm{CD} = \left( \sum_{i=1}^{n} \left( \sum_{j=1}^{m} d_{ij} \right) + D_{is} \right), \tag{3}$$

where "$n$" and "$m$" stand for the number of clusters and related members, respectively; "$d_{ij}$" represents the distance between a node and its CH; and "$D_{is}$" stand for distance between the CH and the BS. The solution suits networks with a large number of widely-spaced nodes. Higher cluster distance leads to higher energy consumption. For minimization of energy consumption, the CD must not be too large [11]. The density of the clusters is controlled by adopting this measurement, while density is the count of nodes in each cluster.

(3) Cluster-based distance-standard CDSD: instead of an average cluster distance, standard derivation measures the changes of distances of the cluster. CDSD is a function of the placement of sensor nodes (random or deterministic). There are clusters with different sizes in random placement so that a SD within a specified variation in the cluster distance is acceptable. If so, the differences in cluster distance is not zero, while the variation must be adopted based on the deployment of information [12]. At any rate, under deterministic placement with uniform distribution of node positions, cluster

distance change must be minimized. Generally, changes of uniform cluster-based distances show that the network is poor, which is not the case when the nodes are placed randomly:

$$\mu = \frac{\sum_{i=1}^{n} d_c}{n},$$
$$\mathrm{SD} = \sqrt{\sum_{i=1}^{n} (\mu - d_c)^2}, \quad (4)$$

"$\mu$" in equations (4) stands for the average of the cluster distances, which is the standard SD formula for obtaining cluster distance variation.

(4) Transfer energy ($E$): it stands for the amount of consumed energy required for transferring all the collected data to the BS. Let "$m$" be the number of associated nodes in a cluster; then, $E$ is obtained by

$$E = \sum_{i=1}^{n} \left( \sum_{j=1}^{m} e_{jm} + m * E_R + e_i \right), \quad (5)$$

where "$e_{jm}$" stands for the required energy to transfer data from a node to the corresponding CH. Thus, the first term in the summation of "$i$" stands for the total consumption of energy for transfer of aggregated data to CHs. Moreover, the second term in the summation "$i$" pictures the total required energy to collect data from members, and finally, "$e_i$" stands for the required energy for transmission from the cluster head to the BS.

(5) Number of transmissions ($T$): in general, the BS dictates the number of transmissions that occurs at every monitoring period. This measure is obtained based on the conditions and the energy level of the network; therefore, "$T$" stands for a long time stage for which the superior optimum solution for maximizing and an inferior solution for minimization are acceptable. The quality of the best solution or chromosome determines the performance of previous GA-based solutions.

In what follows, using genetic algorithm, a fitness function formula to improve each main operational aspects of WSNs (e.g., node placement, network coverage, clustering, and data aggregation) is introduced and discussed. In other words, fitness functions are mainly used to improve energy consumption and lifetime parameters. Simulation results confirmed improvement of the protocols.

## 3. Node Placement in Wireless Sensor Network

The placement of sensor nodes on a monitored field may influence the general performance of the network. Taking into account the placement of nodes in the field, there are three main categories of placement of nodes in a network including the deterministic node placement (grid), the semi-deterministic node placement (e.g., Biased Random), and the nondeterministic (stochastic) node placement (e.g., Simple Diffusion and Random). Long range transmission by sensor nodes is not energy efficient as it needs more energy than a linear function of transmission distance does. Clearly, node density is just one element in network topology as the placement of the node is another key factor. The placement of nodes influences the capacity of a network to correctly sense an event as well as the number of possible disjoint paths towards the sink(s).

Under the deterministic node placement, the nodes are placed on exact, preset points on a grid or in specific parts of the grid. Commonly, deterministic or controlled node placement dictates the type of nodes, the environment that nodes will be placed, and the application. Thus, in Sensor Indoor Surveillance Systems or Building Monitoring application nodes must be placed manually [13]. Under semi-deterministic placement, on the other hand, individual nodes are positioned in a nondeterministic way on the grid (e.g., random) which covers the areas nodes must be spread. That is, microscopic and macroscopic ways of placement of nodes are nondeterministic and deterministic, respectively.

To make sure that network runs with the highest feasible performance, the nodes are positioned on the campus network. Along with balanced energy consumption of all nodes, a preferred node placement protocol is supposed to supply a better network throughput through attenuating contention of channel and collision of packet under high load. An instance of a node placement scheme is pictured in Figure 3(a).

The common advantages of proper sensor propagation in WSNs are listed below [14].

*Scalability.* A high number of nodes can be deployed in the network; this is suitable when transmissions between the nodes are not unlimited.

*Collision Reduction.* Since the cluster head (CH) functions as a coordinator, a limited number of nodes gain access to the channel and cluster members and head communicate locally.

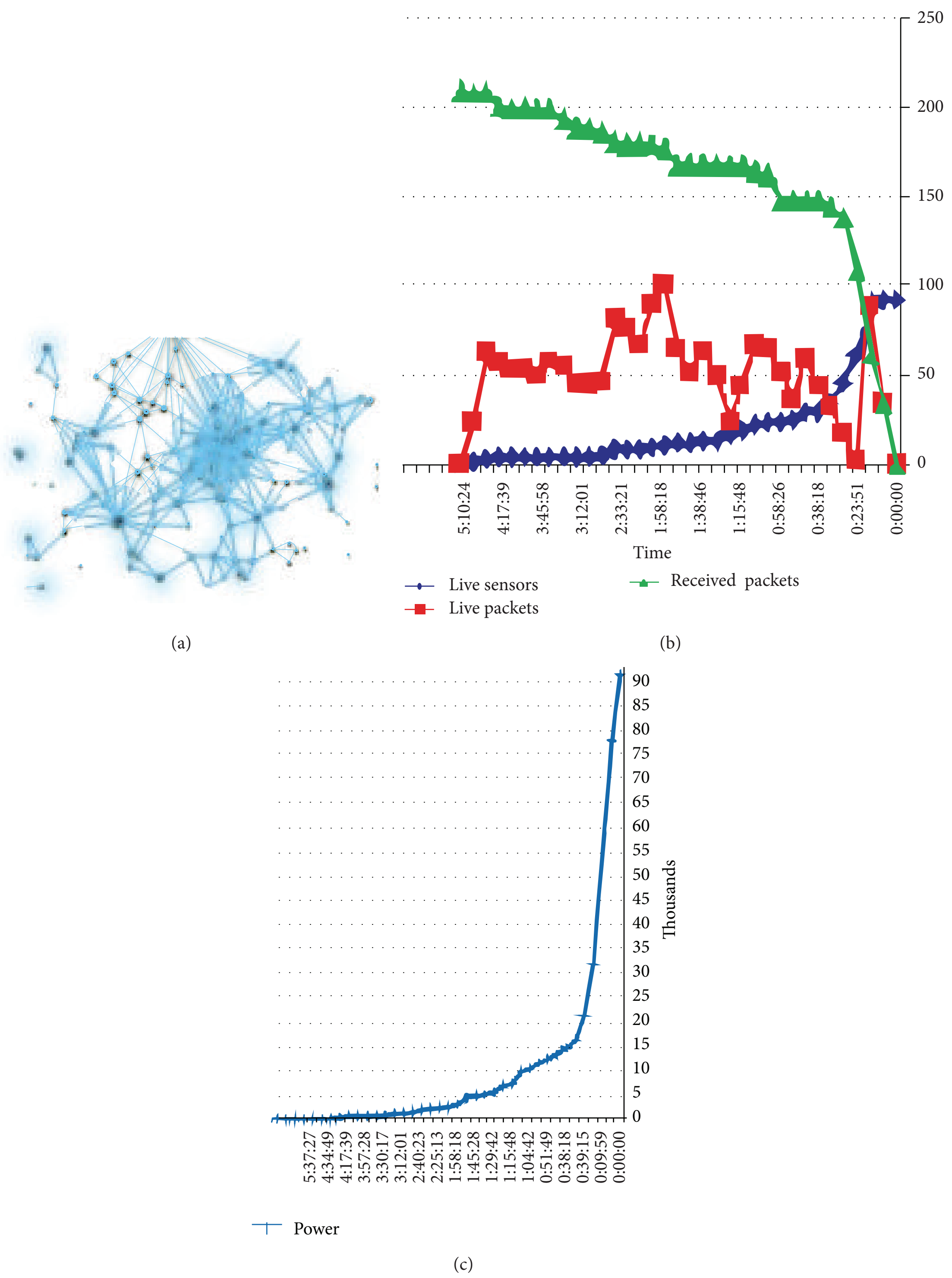
*Energy Efficiency.* High energy consumption is a consequence of the periodic relocation. Still, duties of CH may be distributed among all other nodes through periodic relocation, which results in lower energy consumption.

*Low Cost.* The excess costs can be avoided by deploying sensors at proper locations.

*Routing Backbone.* The data collected by cluster members are aggregated in CH and sent to the sink. Thus, using a little route-thru traffic and routing backbone with enough efficiency one can build the network.

### *3.1. Problem Statement.*

Among the main aspects of improvement of performance for wireless sensor networks, node placement is one to name. Here, we discuss layout optimization of wireless sensor networks (WNNs). All the sensor nodes located in the environment should have a connection with high energy level nodes. For transmitting aggregated data, the nodes relay from environment to base or ground to a satellite. Sensor nodes are not efficient choice for long-term transmission as their energy consumption is a super linear function of the distance the data that is transmitted.



(a)

(b)

(c)

Figure 3: (a) Node placement scheme. (b) Comparison between number of available sensors, live and received packets existing in the network. (c) Comparison between mount of power and lifetime of network.

In this part, we assume that communication range of the sensor is fixed and the new Intelligent Node Placement Protocol in Wireless Sensor Networks using generic algorithm is introduced. The two competing objectives—total sensor coverage and lifetime of the network, are optimized in the proposed framework for WSNs. Thanks to the genetic algorithm, the proposed approach results in a solution where the sensing range is covered with a minimum number of nodes while optimum energy consumption is met.

*3.2. The Proposed Fitness Function.* Calculation of a minimum number of nodes is required in the algorithm. The next step is to evaluate connectivity of the network. This improves architecture of network. In addition, the algorithm takes the connection radius of applied nodes into account. This demonstrates that formula is flexible while different kinds of networks are measured. A fitness function based on the extension of area under coverage is introduced in what follows. It is mainly aimed to realize an optimum solution to cover wider area while efficiency of energy consumption is preserved. Afterward, connectivity of the nodes is examined by prime and Dijkstra algorithms. The coverage and the lifetime of the network are the two main objectives under consideration. The former is obtained by the area of the unit of the disk, which is obtained by radius $R$ centered at each sensor. As per (6), the area of the union is normalized by the total area [15].

$$\text{coverage} = \begin{cases} \bigcup_{i=1}^{n} R^2_{x_i,y_i} & \therefore \left|(x_i, y_i) - (x_{i-1}, y_{i-1})\right| = (0,0) \\ \bigcup_{i=1}^{n} R^2_{x_i,y_i} - r^2_{x_i,y_i} & \therefore o.w., \end{cases} \tag{6}$$

where $\rightarrow\ r_{x_i,y_i} = |(x_i, y_i) - (x_{i-1}, y_{i-1})|$.

Quint et al. introduced the formula (7) for obtaining the energy required for converting a set of purpose points known as purpose function [16]

$$\min\_f = \sum_{i \in s} (\varepsilon + d_i) * y_i + \sum_{i \in D} \mathrm{NC}_j * h_j. \tag{7}$$

In (7), the constant $\varepsilon$ is the required energy to set up a node and $d_i$ represents the cost of routing between upstream through downstream. This mount is computed before launching the program by using the Dijkstra algorithm. This value is a kind of penalty for remote nodes. NC is the surcharge of uncovering points; $y_i$ and $h_j$ show the activation status of node $i$ and covering status of point $j$, respectively. Consider

$$\text{fitness} - \text{function} = \frac{\min\_f_{i,j}}{\text{coverage}_{i,j}}. \tag{8}$$

The proposed fitness function to take into account both the proposed coverage formula and energy procedure is represented in (8). Ration of energy level and amount of energy required to achieve the proper network are obtained by the formula.

*3.3. Evaluation and Simulation Results.* The experiments were conducted with 200 nodes ($N$), a network of $100 * 100\,\text{m}^2$ in extent ($M$), and a BS at 200 m from the network. The simulation parameters are listed in Table 1. As the communication medium between the sensors is binary, optimization of WSN connectivity space is significantly nonlinear. Considerable effects on the two objectives (network disconnection might be the case) is induced by trivial movement of the sensors. Thanks to high efficiency for nonlinear objectives, GA was used in optimization [15].

Table 1: Simulation parameters.

| | |
|---|---|
| Network size | 100 m |
| Node no. | 200 |
| Initial energy | 2 J |
| $E_e$ | 50 nJ/bit |
| $\varepsilon_l$ | 0.0013 pJ/bit/m$^2$ |
| $\varepsilon_s$ | 10 pJ/bit/m$^2$ |
| Network area | $100 * 100\,\text{m}^2$ |
| BS distance | 200 m |
| Packet size | 200 bits |
| $d_{\text{co}} = d_{\text{crossover}}$ | 85 m |

Table 2: GA parameter values.

| | |
|---|---|
| Number of candidate individuals | 100 |
| Length of chromosome | 20 |
| Crossover rate | .5 |
| Mutation rate | .2 |
| Iteration | 100 |

The GA parameters in the environment simulation are listed in Table 2. As it is ineffective on the final results, the chromosomes can be adopted randomly. That is, regardless of chromosome, it tends to the optimum solution. The number of iterations is fixed (100).

The average of experiments on 200 packets is pictured in Figures 3(b) and 3(c). In the latter, the network sustains energy shortage at 0:39:15 due to heavy packets, while the ascending and continuous rate of packet transfer is evident until 5:10:24 in the former. The figures confirm the optimum placement, so that the network tolerates heavy packets optimally. In case of death of one node, the adjacent node is still almost functional for sending the data to the sink. Thus, the proposed algorithm is optimal and improves lifetime of the network [15].

## 4. Network Coverage in WSN

Coverage of WSNs has received great deal of attention in recent researches. The term is usually defined as a measure of performance of lifetime of the sensors in observing the physical space. The coverage is also a critical factor for connectivity of sensor network. By definition, connectivity is the capability of the sensor nodes to communicate with data sink. To deal with the issue of coverage, based on real-world WSN application, a set of hypothetical parameters $(A, B, C)$ were assumed in a 2-D field. Consistent with other studies, the parameters are defined as close as possible to practical situation [12]. As a result, there are three types of sensors monitoring special objectives. For sake of more simplicity of the problem, the presumption is that spatial variables $A$, $B$, and $C$ represent density of sensors per area that monitor objectives in the form of $\rho_A \ll \rho_B \ll \rho_C$ correspondingly. In addition to general aspects of networks, the concept explains specific features of special-objective networks as well.

A Euclidian square field at the length of 1 comprised of identical square area was assumed, so that all the subareas have sensor coverage located at the vicinal intersection lines. The configuration has been adopted in other works as a grid based wireless sensor network layout [12]. Figure 4(a) shows the general scheme of coverage in WSN.

The small sensors are featured with limited-power, limited range of transmission, and sensing mode option (three operating modes) based on capabilities and condition. With lower density of the parameter $A$, sensor is featured with the longest transmission range and $C$ with the shortest range. To achieve optimum energy consumption, a clustering solution, with clusters consisting, one specific adjoining sensor of and the same operating mode known as cluster-in-charge, was devised. All the clusters may use multi-hop to communicate to the base station (BS) or sink. In normal operation, a cluster in charge carries out environment monitoring and data aggregation at specific periodic time and transmits the data to the BS. Here, the multiobjective algorithm capable to optimize the three main parameters (connectivity, consumption of energy, and coverage (ECEP)) of monitoring and measuring at required spots is introduced.

*4.1. Proposed Fitness Function.* To introduce some feasible optimum network topologies with as few as possible constraints (e.g., operational energy, number of unconnected nodes, and cluster-in charge overlap error), a novel algorithm was adopted.

Considering fitness function that takes the whole operational modes in general feasible states, the technique assesses the applied parameters.

*4.1.1. Coverage Problem Formulation to ECEP.* To find the proper fitness functions as a part of genetic algorithm, the formula introduce by Quintão et al. [17] was used; the formula is an improved version of Nakamura's formula [18], where "$A$" is given monitoring area, "$S$" is set of sensor nodes, "$D$" is a set of demanded points, "Ad" is a set of areas needed to be monitored by sensors, "NC" is penalty cost of lack of coverage for the needed point, "AE" is turning energy on, and "PC" is penalty costs of the path stretching from every node to BS (obtained via Dijekstra's algorithm for a processing phase which is dedicated to each node to differentiate expensive nodes). The variables of the model are

$x_{ij} = 1$ when node "$i$" covers demand point $j$ and 0 otherwise,

$y_i = 1$ when nodes "$i$" is active and 0 otherwise,

$h_j = 1$ when demand point "$j$" is not covered.

$$\min \sum_{i\in S} (\mathrm{AE}_i + \mathrm{PC}_i) \times y_i + \sum_{j\in D} \mathrm{NC}_j \times h_j, \tag{9}$$

subject to

$$\begin{gathered} \sum_{ij} \left(x_{ij} + h_j\right) \geq 1, \quad \forall j \in D \;\&\; \forall ij \in A^d, \\ x_{ij} \leq y_i, \quad \forall i \in S \;\&\; \forall ij \in A^d, \\ 0 \leq x_{ij} \leq 1, \quad \forall ij \in A^d, \qquad h_j \geq 0, \quad \forall j \in D, \\ y \in \{0,1\}, \ \{x,h\} \in \mathfrak{R}. \end{gathered} \tag{10}$$

The formula (9) enables us to have minimum required active nodes (more energy to the network) and minimum cases of lack of coverage as well. Every demanded point for monitoring by a sensor or keeping uncovered is represented by constraints (10); they dictate that only active nodes are able of sensing, respectively.

By taking into account penalty cost of overlapping cluster-in-charge errors and consumption of energy marked by OPCE and EC, respectively, an improvement was made in the fitness function (FF). We have

$$\begin{gathered} \mathrm{FF} = \min\left(\mathrm{Usage\_Cost} + \mathrm{Penalty\_Cost}\right), \\ \mathrm{Usage}_{\mathrm{Cost}} = \sum_{i\in s} (\mathrm{AE}_i + \mathrm{PC}_i + \mathrm{EC}) \times y_i, \\ \mathrm{Penalty}_{\mathrm{Cost}} = \sum_{j\in M} \sum_{k\in D} \left(\mathrm{NC}_{jk} + \mathrm{OPCE}_{jk}\right) \times h_k, \end{gathered} \tag{11}$$

subject to

$$M \in \{A, B, C\}. \tag{12}$$

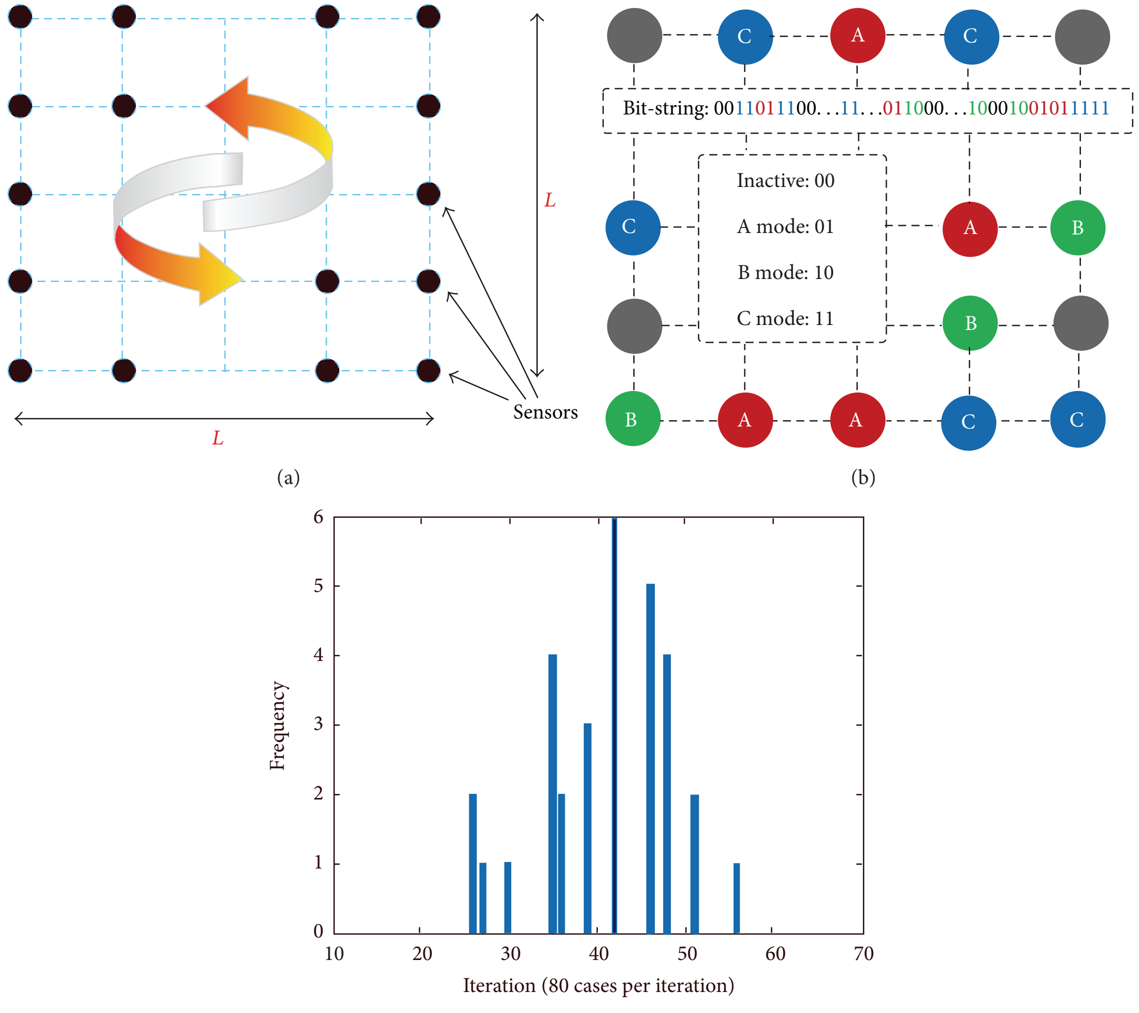
As a dependent to sensor's mode in the network, EC is measured numerically. Clearly, high communication range is obtained by sensor working in mode "$A$" featured with highest rate energy consumption. By assuming 4 and 2 times power usage comparing with "$C$" for mode "$A$" and "$B$", respectively, we have for EC,

$$\mathrm{EC} = \frac{4n_A + 2n_B + n_C}{\sum_{i\in S} n_i}. \tag{13}$$

Taking into account OPCE in FF, wasted energy for overlapping error in cluster-in-charge is obtained.

*4.1.2. Ascertaining Connectivity to ECEP.* Plenty of optimum solutions are obtained by genetic based algorithm, though connectivity of the nodes is not taken into account. This presents outflow of collected data toward the BS. Kruskal algorithm was utilized to examine connectivity of network in the 2nd part of ECEP. The process proposed is comprised of four steps.

The network is assumed as graph $G$ where an edge exists between vertices "$x$" and "$y$" in graph $G$ when the maximum communication range between two particular nodes "$x$" and "$y$" exceeds the distance between "$x$" and "$y$." By introducing the Kruskal algorithm, a minimum spanning tree (MST) is achieved, so that a shorter path between each two vertices for routing aggregated data is achieved.



(a)

(b)

(c)

Figure 4: (a) Coverage scheme in wireless sensor network. (b) Network with represented encoding. (c) Network lifetime in specified scale.

The connectivity with specific shortest paths is achieved when the number of MST tree edges is the same as the number of vertices –1; otherwise, inactive nodes are activated (this explains shorter transmission range than communication distance for some nodes). Kruskal technique is used on newly activated and disconnected nodes. This results in formation of new lightest tree.

The shortest path between each disconnected node to the BS is obtained and the internal sensor nodes of the paths are added to the set $E$.

Any newly activated node not listed in $E$ is turned deactivated. This helps preservation of network energy while the quality is the same.

Finally, one or two network typology(s) were developed based on the range of transmission of nodes and position of sensor node. The networks that realized maximum network coverage are characterized with optimum coverage and energy usage.

*4.1.3. Encoding.* The proposed approach was implemented in a square field ($L * L$) with virtually equal subareas. Each node is positioned at intersection of the subareas and obtains four expressions: (1) active (00), (2) mode $A$ active (01), (3) mode $B$ active (10), and (4) mode $C$ active (11). Figure 4(b) shows the network with encoding.

The whole nodes arrangement in the network resembles a chromosome. That is, each node represents gen and a set of gens in specific order creates a certain chromosome. In this way, $L$ nodes in a network host $2L2$ bits as mode of each node is showed by two bits and there are $L2$ gens in the network.

*4.2. Evaluation and Simulation Results.* Genetic algorithm technique includes set of chromosomes known as population which improves by generation process. To put it another way, inspired by the nature, the algorithm receives input data which are randomly collected by the primary population. When generation process is completed, the final population/result represents the optimum solution for the main problem. In general, all improvements made by the generation process are comprised of crossover, scoring, selection, and mutation function. The term crossover refers to productive function at specific rate where two different chromosomes mate to produce new generation. Among different methods of crossover, single point is under focus here.

As the most critical part of genetic algorithm, scoring or assigning fitness, on the other hand, employs FF for scoring. Specific weight is assigned to each chromosome depending on the content. This is to say that each chromosome is a solution developed through iterations. There is a direct relation between fitness value of chromosome and chance of surviving in some generations. The FF is a totally problem-based design and achieving intelligent fitness function to differentiate qualified people has been the main concern of the literature. Superior chromosomes are adopted by selection process to create a new population with mutation technique that permits specific chromosomes to enter the new generation. The stochastic nature of GA dictates that different solutions with variant performance are obtained in different runs of the algorithm. The proposed algorithm was implemented by WSN simulator and almost 100% coverage over the monitored area was realized.

Level of energy consumption and number of active nodes along with live packets over time are listed in Table 3. Only 18 packets were first delivered to BS and later the number increased to 83 packets in second 01:18.

Table 4 represents that the network died last time in 5:13:781. Moreover, there is a gradual decrease in number of live packets as number of active sensor converges to 0.

Role of number of nodes on genetic algorithm iteration on lifetime of the network is pictured in Figure 4(c). It is implied by iterations 50 and 57 that more number of individuals does not necessary result in better solution. That is, meeting stop criterion is enough to reach the optimum solution.

The results of simulation confirmed merits of relatively large number of sensors with low energy consumption over activating fewer numbers of sensors with higher energy consumption.

Table 3: Observed values in the early times of network.

| Time (nanosecond) | Power | Active sensors/35 | Live packets |
|---|---|---|---|
| 00:35.090 | 20718 | 27 | 18 |
| 00:49.330 | 15332 | 27 | 60 |
| 01:08.107 | 8031 | 20 | 53 |
| 01:18.182 | 6739 | 18 | 83 |
| 01:36.719 | 5333 | 14 | 30 |
| 01:48.626 | 4368 | 12 | 36 |
| 02:02.956 | 3354 | 10 | 38 |
| 02:09.666 | 3123 | 8 | 38 |
| 02:18.048 | 2897 | 8 | 47 |
| 02:46.709 | 2032 | 7 | 54 |
| 02:55.352 | 1860 | 6 | 52 |

Table 4: Observed values in the last times of network.

| Time (nanosecond) | Power | Active sensors/35 | Live packets |
|---|---|---|---|
| 03:22.821 | 1096 | 5 | 87 |
| 03:45.804 | 616 | 4 | 89 |
| 03:58.382 | 421 | 3 | 75 |
| 04:15.196 | 307 | 2 | 33 |
| 04:39.882 | 122 | 1 | 25 |
| 05:13.781 | 0 | 0 | 0 |

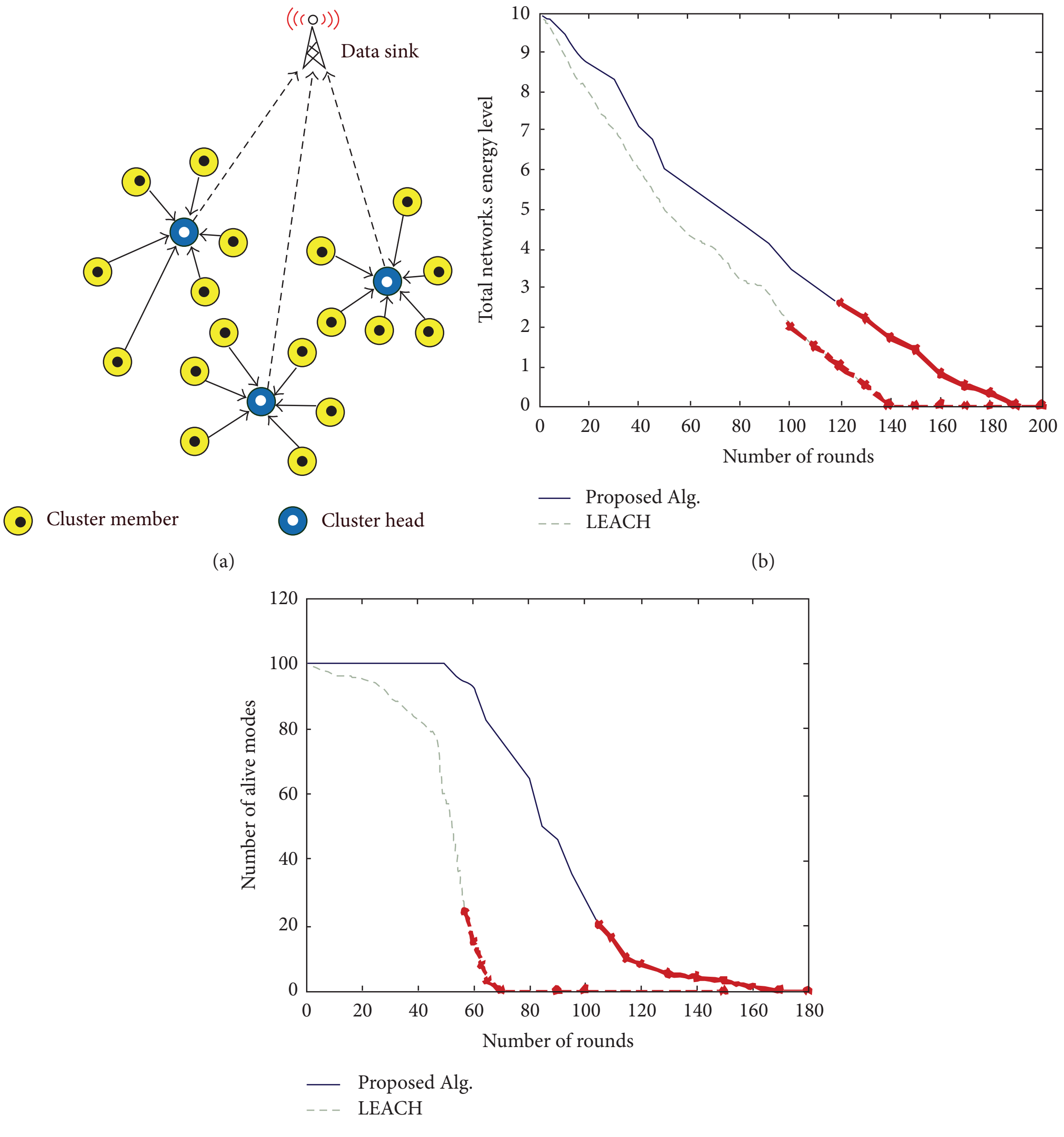
## 5. Clustering in WSN

As mentioned earlier, increase of lifetime and expansion and load balance are the main requirements of WSNs applications. Proper clustering using optimized techniques of clustering is an option to realize these goals.

Generally, the cluster based methods suit monitoring applications featured with necessity of nonstop stream of data from sensors [11]; this calls for reducing the costs of timely data message delivery by routing protocols. The Low Energy Adaptive Clustering Hierarchy (LEACH) protocol, for instance, employs a hierarchical approach for clustering the network. There is an adopted cluster head for managing each cluster. The cluster head is in charge of several tasks; first it is comprised of collecting data supplied from the members of a cluster on periodical basis. It aggregates the data after gathering them, so that redundancy among correlated values is dealt with [19]. The next main task assigned to a cluster head is to directly transmit the aggregated data to the base station. The transmission is conducted via a single hop. Creating a TDMA-based schedule is aimed to assign a time slot to each cluster—to be used for transmission is the main task. The cluster members learn by the schedule when the head cluster disseminates it. To minimize probability of collision among the sensors inside and outside a cluster, a code-division multiple access-based scheme is employed by LEACH nodes for communication.

Figure 5(a) illustrates a sample of cluster based WSN. The aggregated data are transmitted from several clusters via CHs toward BS. The CHs are marked by gray circle. There are plenty of studies on protocols with less costs of transmission between CHs and BS.

*5.1. The Proposed Fitness Function.* For defining energy consumption and improve lifetime of the network, the parameters of the genetic algorithm were set according to software services. There is a negative relation between energy consumption and distance parameters. One way to lessen the distance between member nodes and pertinent CH is to use more clusters; each cluster may have one or more cluster head(s), which is not economic regarding the energy consumption. However, by using more clusters we avoid longer distances. Because of this, to achieve average amount of energy consumption by each node, a ratio of total energy usage to the total distances of nodes was defined. We propose a formula to achieve optimal WSN energy consumption and coverage (14).



(a)

(b)

(c)

Figure 5: (a) A sample of cluster based WSN. (b) Energy consumption rate over the lifetime of a network. (c) Comparison of live nodes in two methods.

Where, $((e_i * T) \times e_j * T)$ is the total energy consumption and $((D_a \times \text{nodes}) \times (D_b \times \text{CHs}))$ is the total distance between nodes and each cluster is multiplied by total distance between cluster heads. $F(i)$ represents the maximum achievable value for the ratio. Taking into account the negative relation between number of cluster heads and number of nodes/CHs and the same relation between $e_k$ and $e_j$ on one hand and amount of $D_a$ and $D_b$ on the other hand, the maximum value of ratio action is a trade-off of energy consumption and number of blusters [17]:

$$F(i) = \left(\frac{e_i \times T}{D_a \times \#\text{Nodes}}\right) \times \left(\frac{e_j \times T}{D_b \times \#\text{CHs}}\right), \tag{14}$$

$$\begin{gathered} \therefore D_a = \frac{\text{Width} \times g_i}{\sqrt{\#\text{Clusters}}} \\ \forall g_i \in \{\text{DDBS}, \text{CD}, \text{CDSD}\} \\ [g_i = \text{DDBS}] = 1 \because [\#\text{clusters} = 1] = 1 \\ \therefore \left(D_a = D_b = \#\text{CHs} = 1\right). \end{gathered} \tag{15}$$

$F(i)$ in the intelligent suitability function is capable of grading any chromosome both through cluster based method or direct method. To achieve the optimum solution, the optimum chromosome choice is made based on passing generation.

*5.2. Evaluation and Simulation Result.* A comparison is made between the GA-based approach proposed here and other cluster-based protocols (e.g., LEACH).

The parameters used in the simulation are listed in Table 5 and the clusters are featured with only one CH, while the generic algorithm process is used to obtain the number of CHs.

Table 6 represents the parameters of GA in the simulation. It is possible to adopt the chromosomes through random selection. The number of iterations is constant at 100.

A comparison between the proposed algorithm and LEACH regarding network energy and lifetime is pictured in Figures 5(b) and 5(c). The comparison is made over 200 periods of time. The former represents that unified consumption of energy by CHs results in short lifetime of nodes in LEACH. The latter shows that death of the first node in the proposed algorithm is delayed compared with LEACH as the node is removed due to energy status. Moreover, the network keeps working with minimum number of alive nodes. In general, the final individuals form a cluster with uniform energy consumption. This happens thanks to algorithm fitness function that takes the energy status of nodes and CHs/BS distances. In this way, the phenomenon adds to lifetime of the network significantly [20].

Table 5: Simulation parameters.

| | |
|---|---|
| Network size | 100 m |
| Node no. | 200 |
| Initial energy | 2 J |
| $E_e$ | 50 nJ/bit |
| $\varepsilon_l$ | 0.0013 pJ/bit/m$^2$ |
| $\varepsilon_s$ | 10 pJ/bit/m$^2$ |
| Network area | 100 $*$ 100 m$^2$ |
| BS distance | 200 m |
| Packet size | 200 bits |
| $d_{\text{co}} = d_{\text{crossover}}$ | 85 m |

Table 6: GA parameter values.

| | |
|---|---|
| Number of candidate individuals | 100 |
| Length of chromosome | 20 |
| Crossover rate | .5 |
| Mutation rate | .2 |
| Iteration | 100 |

## 6. Data Aggregation in WSN

The purpose of data aggregation is to collect the highly critical data supplied by the sensors and to forward the data to the sink. Efficient energy consumption and reducing data latency as much as possible are two main concerns. The latter is vital for many applications including environment monitoring where fresh data are imperative. Achieving higher energy efficiency in data aggregation algorithm ensures longer network lifetime. Failing to share the load of data among the members of a network by the data aggregation tree eventuates in consumption of total energy by some of the nodes that are assigned with heavy load of data. Failure of nodes leads to failure of the network. Utilizing GA, this section investigates the data collecting spanning trees with higher energy efficiency. We try to achieve a proper route that balances the data load over the network. An algorithm that ensures a balance of residual energy among the nodes increases lifetime of the network.

The highest distance between every pair of nodes of the two clusters determines the distance between the clusters. MLDA is utilized on the basis of this cluster information. The (EESR) Energy-Efficient Sensor Routing was introduced by Hussain and Islam to be used on multi-hop network. To have higher efficiency of energy consumption, they used a spanning tree which is in fact a group of routing trees [21]. According to EESR, for calculating energy edge cost, the node with minimum energy is adopted. It uses the node with minimum energy and takes into account the lowest and highest cost links that receives data packets from the neighbor node and forward them to BS. Two spanning trees, featured with aggregation scheme and data gathering designed to guarantee higher lifetime of the network, were studied by Tan and körpeoğlu [22].

Yang and Fei proposed a new approach called Intermediate Target Based Geographic Routing (ITGR) to avoid such long detour paths. The novelty of the approach is that a single forwarding path can be used to determine a shaded area that may cover many destination nodes. They designed an efficient method for the source to find out whether a destination node belongs to a shaded area [23].

In general, there are two methods for power management among the nodes in data aggregation stage. One is the power aware version (PEDAPPA) that tries to achieve higher lifetime through creating balanced energy consumption by the nodes. The second method (PEDAP), the nonpower aware version, on the other hand, tries to attenuate energy consumption by the system on the basis of data gathering round [24]. The method ensures higher lifetime of the last node. Each method adopts different approaches to calculate the edge cost.

*6.1. Problem Statement.* The first assumption is that the network is initialized with every node having a fix range of radio communication and a specific primary energy before receiving the multi-data packet. All nodes are capable to monitor the environment, to send children packets to the neighbors, and to send single one to the corresponding parents. This process is performed as long as possible. According to the proposed algorithm in this part, every node, after initialization, may send a sample certain packet to the BS. In case a route is adopted for transmitting data packet, the BS utilizes a routing table with all the properties and current energy of the nodes listed to search for the monitor node with corresponding property and sends the resultant information to GA.

To find out an optimum route, the GA generates all possible routes. Afterward, BS prepares a schedule based on the route and sends it to all nodes. Then, the routing table is updated once more by GA while applying the reduction of energy for all the nodes. In effect, GA minimum spanning tree and aggregation tree are alike as the former is based on the environment-monitor node developed to examine the best edges toward the BS and to achieve balanced load of data packets to the nodes. In this case, the network is considered alive as far as the minimum required nodes are active to send data packet.

*6.2. Fitness Function.* Under fitness function, every chromosome is scored. This lets us to make comparison regarding number of deaths or survival over all the members. A formula later improved to be known as Nakamura formula was used in development of our proposed fitness function. Under the formula, "$A$" stands for the given monitoring area, "$S$" stands for the set of sensor nodes, "$D$" stands for the group of demanded points, "Ad" stands for the set of sensors in charge of monitoring the area under consideration, "AL" stands for the turning energy on in low energy level mode, "AH" stands for turning energy on in high energy level mode (sending packet load to the just "BS" stands, "EC" stands for the cost of a node in 3 states ($A$, $B$, $C$), and "$S$" stands for the set of edge collection Fitness function that is a procedure which scores any chromosome. This value helps us to compare the whole ones to each other for survival or death (15). The model can be formulated as

$$G(i) = \sum_{i \in S} (\mathrm{AE}_i + \mathrm{EC}_i) \quad \forall i \in D \,\&\, \forall ii \in A^d, \qquad \mathrm{AE} \in \{\mathrm{AL}, \mathrm{AH}\}. \tag{16}$$

The formula above (16) obtains all the feasible paths and the nodes' status. This information is required to set a node in low/high level energy mode. The proposed fitness function was improved by defining at least 3 states for all the node for obtaining the almost exact fitness as

$$\mathrm{EC} = \frac{4n_A + 2n_B + n_C}{\sum_{i \in S} n_i} \qquad M \in \{A, B, C\}. \tag{17}$$

The mode of the sensor network affects the EC, which is measured numerically. Knowing high range of communication for the sensor node in mode $A$, the highest energy consumption is expected under this mode. Modes $B$ and $C$, on the other hand, with shorter communication range are next in order of energy consumption. It is assumed that energy consumption under mode $A$ is four times more than that of $C$, and that of $B$ in turn is 2 times more than that of $C$. EC is obtained through [25].

In the fitness function below, $N$ stands for number of nodes; the function yields the electrical power based on setup energy. The energy needed for sending data packet from the children is marked by $E_{\mathrm{children}}$:

$$E(i)_{\mathrm{Total}} = E_{\mathrm{Monitor}} + G(i), \qquad F(i) = \frac{E(i)_{\mathrm{Total}}}{N}. \tag{18}$$

The formula above gives the average energy mount through dividing by the number of nodes. The selection function is used to assess each individual, so that the better the fitness value, the more the chance for surviving to the next generation.

Table 7: Simulation parameters.

| | |
|---|---|
| Network size | 100 m |
| Node no. | 200 |
| Initial energy | 0.8 J |
| BS location | Center of resource |
| Network area | 100 * 100 m$^2$ |
| Scenario simulated | 5 times that average one is reported |
| Tree used time | 15 periods |

Table 8: GA parameter values.

| | |
|---|---|
| Number of candidate individuals | 500 |
| Length of chromosome | 20 |
| Crossover rate | .7 |
| Mutation rate | .7 |
| Iteration | 200 |

*6.3. Evaluation and Simulation Results.* A network simulator was used for implementation of the algorithm. The simulation is featured with two steps; first, Java editor is utilized in implementation of genetic algorithm based portion. To this end, Java Genetic Algorithm package (JPAC) was installed—there were other studies using the same method. Afterward, OMNET++ was employed to track the different routes between sensor node and BS in some simulated environment. Table 7 lists the parameters for the experienced sensor network.

As Table 7 represents, each tree in the simulation is employed for 15 periods and the simulation is repeated 5 times for each scenario to obtain an average to report.

Parameters of the simulation of the environment are listed in Table 8. Taking into account priorities such as remaining energy and packet size, the simulation found all routes and GA adopted the optimum one out of them. Afterward, all nodes in the selected routes were scheduled by BS. Comparing with the other studies, the simulation showed that the proposed fitness function met the objectives.

Comparisons between the proposed algorithm here and LEACH protocol on network energy and lifetime of 200 periods of time (year) are illustrated in Figures 6(a) and 6(b). Figure 6(a) shows the unified packet load and energy consumption for obtaining the optimum route toward BS. On the other hand, Figure 6(b) pictures the time and the first node is removed. Clearly, in comparison with LEACH protocol, death time of the first node is considerably delayed. In addition, the network can keep working with minimum

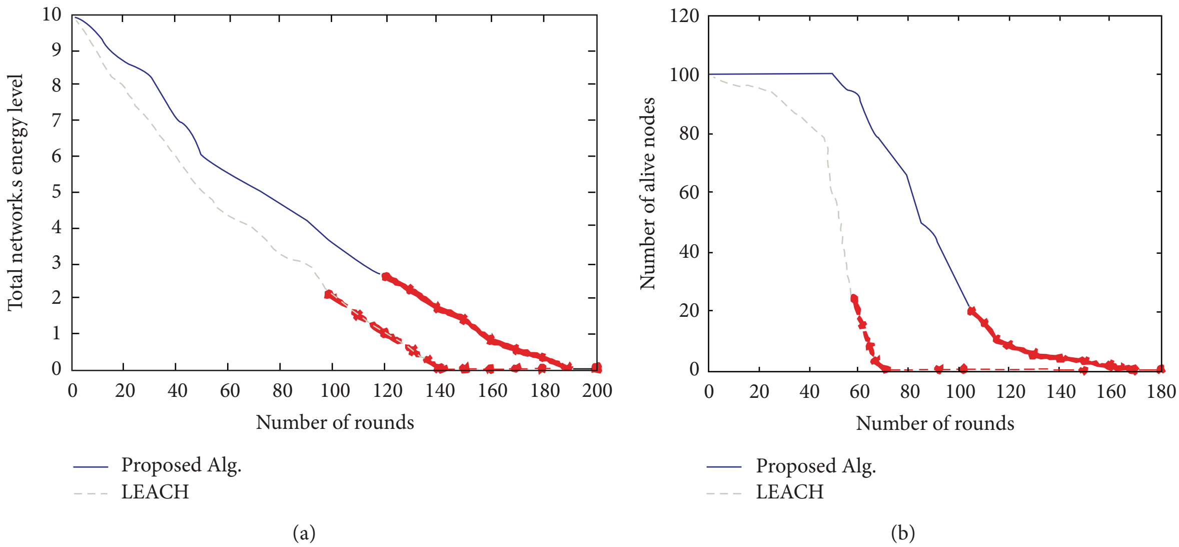


Figure 6: (a) Energy consumption rate in lifetime of virtual environments. (b) Comparison of coverage in two methods.

number of active node. In general, by using routing algorithm, fitness function takes three states of energy status (low, middle, and high) or communication range for the BS. The final individuals provide route with almost uniform energy consumption. This feature makes a great contribution to the network's lifetime [25].

## 7. Conclusion and Future Work

WSNs are comprised of a set of wireless sensors with variety of capabilities and limitations, which make them suitable for specific applications. There are several imaginable applications for WSNs in military, commercial, and medical fields. Taking into consideration the recent technological advances, utilization of these networks in daily life is increasing. Of the main limitations of WSNs is energy consumption and lifetime of the network, which are common concerns almost for any WSN application. In general, the operational stages of WSNs include node placement, network coverage, clustering, data aggregation, and routing. A technical survey was conducted on these operational stages. By finding the drawbacks and optimizing them, ideal parameters of the network were achieved. Finally, using genetic algorithm, a fitness function with optimum formula was obtained and the present protocols were optimized. The results of simulations in JPAC, MATLAB, and NS were compared with are of the present protocols and optimization of the two parameters confirmed. It is also noticeable that the diagrams obtained from the simulations showed an improvement in energy consumption parameters and lifetime of the network; this means more ideal WSNs. An application based protocol without specific limitation regarding its application—suitable for military, medical, and commercial applications—will be subject of our future studies.

## Conflict of Interests

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

## References

[1] I. F. Akyildiz and M. C. Vuran, "Front matter," in *Wireless Sensor Networks*, John Wiley & Sons, Chichester, UK, 2010.

[2] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[3] F. S. Babamir and A. Norouzi, "Achieving key privacy and invisibility for unattended wireless sensor networks in healthcare," *The Computer Journal*, 2013.

[4] F. S. Babamir and Z. Eslami, "Data security in unattended wireless sensor networks through aggregate signcryption," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 11, pp. 2940–2955, 2012.

[5] M. Romoozi, M. Vahidipour, M. Romoozi, and S. Maghsoodi, "Genetic algorithm for energy efficient & coverage-preserved positioning in wireless sensor networks," in *Proceedings of the IEEE International Conference on Intelligent Computing and Cognitive Informatics (ICICCI '10)*, pp. 22–25, June 2010.

[6] A. Norouzi, A. H. Zaim, and A. Sertbas, "A comparative study based on power usage performance for routing protocols in wireless sensor network," in *Proceedings of the IEEE International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE '13)*, Konya, Turkey, May 2013.

[7] A. Norouzi, F. S. Babamir, and A. H. Zaim, "An interactive genetic algorithm for mobile sensor networks," *Studies in Informatics and Control*, vol. 22, no. 2, pp. 213–218, 2013.

[8] D. E. Goldberg, *Genetic Algorithm in a Search Optimization and Machine Learning*, Addison Wesley, 1989.

[9] V. Kreinovich, C. Quintana, and O. Fuentes, "Genetic algorithms. What fitness scaling is optimal?" *Cybernetics and Systems*, vol. 24, no. 1, pp. 9–26, 1993.

[10] A. Norouzi, F. S. Babamir, and A. H. Zaim, "A novel energy efficient routing protocol in wireless sensor networks," *Journal of Wireless Sensor Network*, vol. 3, no. 10, pp. 1–10, 2011.

[11] S. Hussain, A. Matin, and O. Islam, "Genetic algorithm for hierarchical wireless sensor network," *Journal of Networks*, vol. 2, no. 5, pp. 87–97, 2007.

[12] A. P. Bhondekar, R. Vig, M. L. Singla, C. Ghanshyam, and P. Kapur, "Genetic algorithm based node placement methodology for wireless sensor networks," in *Proceedings of the International MultiConference of Engineering and Computer Science (IMECS '09)*, vol. 1, pp. 106–112, Hong Kong, China, March 2009.

[13] C. Sergiou and V. Vassiliou, "Chapter 1: efficient node placement for congestion control in wireless sensor networks," in *Wireless Sensor Networks: Technology and Applications*, InTech.

[14] N. Trigoni, Y. Yao, A. Demers, J. Gehrke, and R. Rajaramany, "Wave scheduling: energy-efficient data dissemination for sensor networks," in *Proceedings of the International Workshop on Data Management for Sensor Networks (DMSN), in Conjunction with the International Confernece on Very Large Data Bases (VLDB)*, August 2004.

[15] Z. Orman, A. Norouzi, and F. S. Babamir, "Intelligent node placement using GA (INPGA) protocol in wireless sensor networks," in *Proceedings of the International Conference on Wireless Networks (ICWN'11)*, 2011.

[16] F. P. Quint, F. Nakamural, and G. R. Mateusl, "A Hybrid approach to solve the coverage and connectivity problem in wireless sensor networks," in *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Ill, USA, 2005.

[17] F. P. Quintão, F.G. Nakamura, and G. R. Mateus, "A hybrid approach to solve the coverage and connectivity problem in wireless sensor networks," in *Proceedings of the 4th European Workshop on Meta-Heuristics: Design and Evaluation of Advanced Hybrid Meta-Heuristics*, Nottingham, UK, 2004.

[18] F. G. Nakamura, *Planejamento Din.amico para Controle de Cobertura e Conectividade em Redes de Sensores sem fio planas. [Master's thesis]*, Universidade Federal de Minas Gerais, 2003, (portuguese).

[19] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*, John Wiley & Sons, 2007.

[20] A. Norouzi, F. Babamir, and A. Zaim, "A new clustering protocol for wireless sensor networks using genetic algorithm approach," *Wireless Sensor Network*, vol. 3, no. 11, pp. 362–370, 2011.

[21] S. Hussain and O. Islam, "An energy efficient spanning tree based multi-hop routing in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 4386–4391, March 2007.

[22] H. Ö. Tan and I. Körpeoğlu, "Power efficient data gathering and aggregation in wireless sensor networks," *SIGMOD Record*, vol. 32, no. 4, pp. 66–71, 2003.

[23] J. Yang and Z. Fei, "ITGR: intermediate target based geographic routing," in *Proceedings of the 19th International Conference on Computer Communications and Networks (ICCCN '10)*, pp. 1–6, IEEE, Zurich, Switzerland, August 2010.

[24] O. Islam, S. Hussain, and H. Zhang, "Genetic algorithm for data aggregation trees in wireless sensor networks," in *Proceedings of the 3rd IET International Conference on Intelligent Environments (IE '07)*, pp. 312–316, September 2007.

[25] A. Norouzi, F. Babamir, and Z. Orman, "A tree based data aggregation scheme for wireless sensor networks using GA," *Wireless Sensor Network*, vol. 4, no. 8, pp. 191–196, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 354857, 7 pages
http://dx.doi.org/10.1155/2014/354857

*Research Article*

# Multiobjective Robust Design of the Double Wishbone Suspension System Based on Particle Swarm Optimization

**Xianfu Cheng and Yuqun Lin**

*School of Mechanical and Electronical Engineering, East China Jiaotong University, Nanchang 330013, China*

Correspondence should be addressed to Xianfu Cheng; chxf_xn@sina.com

Received 22 October 2013; Accepted 19 December 2013; Published 11 February 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

The performance of the suspension system is one of the most important factors in the vehicle design. For the double wishbone suspension system, the conventional deterministic optimization does not consider any deviations of design parameters, so design sensitivity analysis and robust optimization design are proposed. In this study, the design parameters of the robust optimization are the positions of the key points, and the random factors are the uncertainties in manufacturing. A simplified model of the double wishbone suspension is established by software ADAMS. The sensitivity analysis is utilized to determine main design variables. Then, the simulation experiment is arranged and the Latin hypercube design is adopted to find the initial points. The Kriging model is employed for fitting the mean and variance of the quality characteristics according to the simulation results. Further, a particle swarm optimization method based on simple PSO is applied and the tradeoff between the mean and deviation of performance is made to solve the robust optimization problem of the double wishbone suspension system.

## 1. Introduction

Suspension used in an automobile is a system mediating the interface between the vehicle and the road, and the functions of it are related to a wide range of drivability such as handing ability, stability, and comfortability [1]. There are many different structures of vehicle suspension system according to the mechanical jointing pattern, the type of springs, the independence of the left and right wheels, and so forth, of which the independent double wishbone suspension is extensively used.

With reference to automobile suspension system, a number of researches have devoted considerable efforts to design optimization. Many important relationships have been highlighted among vehicle suspension parameters and suspension performance indices [2]. These researches can be classed into several aspects: (1) a single-objective optimization of separately only considering reducing the dynamic load of the tire on the road or smoothness [3]; (2) transforming the traditional multiobjective optimization problem into single-objective optimization problem through a mathematical transformation [4]; (3) using multiobjective and multidecision optimization of true sense of the decision making after the first optimization [5]; (4) carrying out the analyses of displacement, velocity, and acceleration for McPherson strut suspension system using displacement matrix [6]. The optimal design is a balance of the kinematics and compliance characteristics of the suspension system [7]. But these approaches are based on conventional deterministic optimization and do not consider any deviations of design parameters, such as manufacturing errors of parts, which may result in unreliability of design objectives and constraints, and the computation time of these approaches is enormous. Robust design is powerful and effective in helping manufacturers to design their products and process as well as to solve troublesome quality problems, ultimately leading to higher customer satisfaction and operational performance [7]. However, a comprehensive multi-objective and robust approach seems to little be addressed. Chun et al. studied optimal designs for suspension systems based on reliability analyses [8]. Choi et al. performed a reliability optimization with the single-loop single-variable method by using results

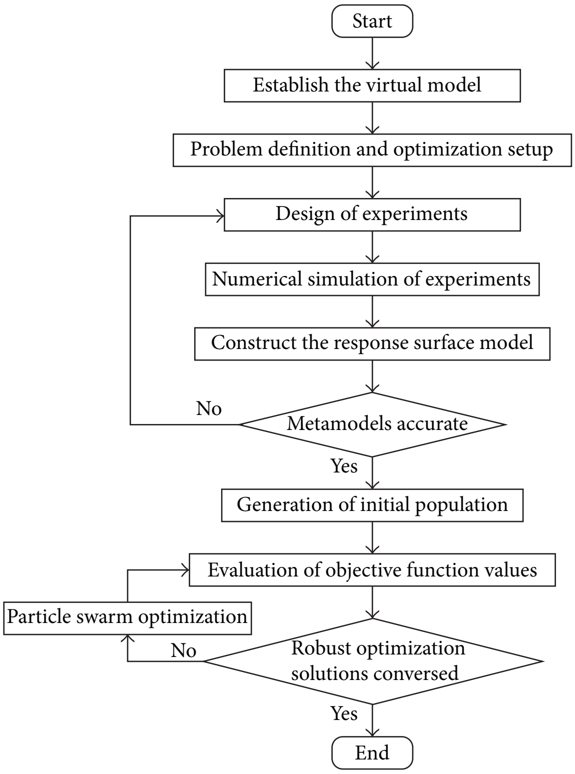

Start
Establish the virtual model
Problem definition and optimization setup
Design of experiments
Numerical simulation of experiments
Construct the response surface model
No
Metamodels accurate
Yes
Generation of initial population
Evaluation of objective function values
Particle swarm optimization
No
Robust optimization solutions conversed
Yes
End

Figure 1: Robust design based on particle swarm optimization.

of a deterministic optimization as initial values of reliability-based optimization using the finite difference design sensitivity [9].

Robust optimization design is essentially multiple objectives: (1) optimizing the mean of performance and (2) minimizing the variation of performance. Since performance variation is often minimized at the cost of sacrificing performance, a tradeoff between the aforementioned two aims is generally presented. The particle swarm optimization (PSO) approach has demonstrated its strength in various types of multiobjective optimization design, including vehicles, aircrafts, and manufacturing facilities [10]. So, this study presents a new optimal method—based on robust design and PSO for suspension system. The objectives are the toe angle and the lateral slip of the wheel grounding point and their variations of the double wishbone suspension system. The mean and variance models are established by the Kriging model, and then PSO is used to analyze the robust performance of the system. This may help the designers to identify layout of the suspension system and to develop the optimum design system of suspension.

This paper is organized as follows: in Section 2, the virtual model of the double wishbone suspension system is established; in Section 3, the model of multi-objective robust optimization is built; the robust designs based on particle swarm optimization are described and analyzed in Section 4 and conclusions are presented in Section 5. The process of the robust design based on particle swarm optimization is shown in Figure 1.

Table 1: Key point.

| Key point | $x$ | $y$ | $z$ |
|---|---|---|---|
| Lca front | −200 | −400 | 150 |
| Lca back | 200 | −450 | 155 |
| Lca outer | 0 | −750 | 100 |
| Uca front | 100 | −450 | 525 |
| Uca back | 250 | −490 | 530 |
| Uca outer | 40 | −675 | 525 |
| Shaft inner | 0 | −200 | 225 |
| Spring lower | 0 | −600 | 150 |
| Subframe front | −400 | −450 | 150 |
| Subframe back | 400 | −450 | 150 |
| Tie rod inner | 200 | −400 | 300 |
| Tie rod outer | 150 | −750 | 300 |
| Spring upper | 40 | −500 | 650 |
| The center of the wheel | 0 | −800 | 300 |

## 2. The Virtual Model of the Double Wishbone Suspension System

The double-wishbone suspension system is a group of space RSSR (revolute joint—spherical join—spherical join—revolute joint) four-bar linkage mechanisms. Its kinematics relations are complicated, kinematics visualization analysis is difficult, and its performance is poor. Thus, rational settings of the position parameters of the guiding mechanism are crucial to assuring good performance of the independent double-wishbone suspension.

The vehicle's right and left suspensions are symmetrical, so choose the left or the right part of the suspension system which is studied to simulate the entire mechanism, excluding the variation of wheel centre distance (WCD) which is advisable. The key design parameters are the coordinates of the key points (see Table 1) and the assembly relationship between every member. A model of the left half of an independent double wishbone suspension system is established, as shown in Figure 2. Major components include the upper control arm (UCA), lower control arm (LCA), tie rod, knuckle, spring, and absorber. The design purpose of this study is to determine the positions of the joints. A commercial program, ADAMS, is employed for modelling and analysing the suspension system.

Make the following assumptions on double wishbone suspension.

(1) The composition members of the suspension are rigid body, and the elastic deformation is ignored.

(2) Rigid connection between the various components is used and ignores the internal clearance and friction.

(3) Only consider the ground roughness, without regarding to the dynamic factors.

(4) Add an incentive on the test platform to simulate the unevenness of the ground; the tires are always in contact with the test bench.

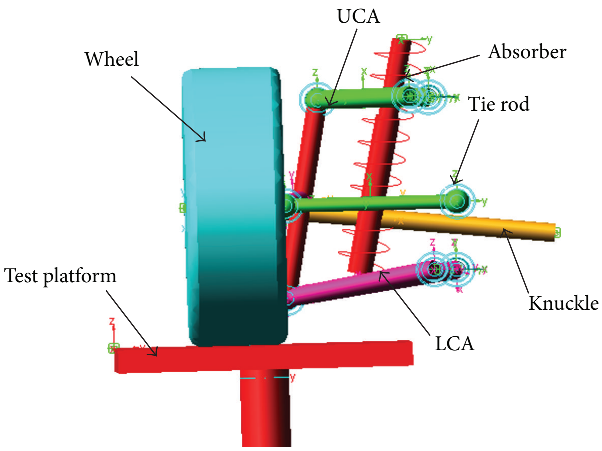


Figure 2: Double wishbone suspension model.

Add an excitation source on the test platform, $y = 50\sin(2\pi t)$, and then taking numerical simulation, the results are shown in Figures 3 and 4.

As shown in Figure 3, the wheel sideways displacement changes with time. The change of sideways displacement is calculated according to the variation of the wheel travel. As shown in Figure 4, the toe angle changes with time. The change of toe angle is calculated according to the variation of the wheel travel too.

## 3. Model of the Robust Design and Approximation Model

In this section, the model of the robust design is built. A full-factor test and sensitivity analysis are utilized to determine main design variables. The Latin hypercube design is adopted to find the initial point, and the database is created for fitting the kriging model of the robust design.

*3.1. Robust Design.* Robust design has become a powerful tool to aid designers in making judicious selection and control of variation. The fundamental principle of robust design is to improve the quality of a product by eliminating the variation of controllable factors (i.e., dimension, assemble gap, material properties, etc.) and uncontrollable factors (i.e., applied loadings, environment, aging, etc.). Consequently, compared with traditional optimization design, robust design can make the product maintain good performance [11].

A standard engineering optimization problem is normally formulated as follows:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_j(x) \le 0, \quad j = 1, 2, \ldots J \\ & x_L < x < x_U, \end{aligned} \tag{1}$$

where $f(x)$ is the objective function and $g_j(x)$ is the $j$th constraint function; $x$, $x_L$, and $x_U$ are vectors of design variables, their lower bounds, and upper bounds, respectively. If the design variable $x$ follows a statistical distribution, a robust design problem can be stated as a biobjective robust design problem as follows:

$$\begin{aligned} \min \quad & \left[\mu_f, \sigma_f\right] \\ \text{s.t.} \quad & g_j(x) + k_j \sum_{i=1}^{n} \left| \frac{\partial_{g_j}}{\partial_{x_i}} \right| \Delta x_i, \quad j = 1, \ldots, J \\ & x_L + \Delta x \le x \le x_U - \Delta x, \end{aligned} \tag{2}$$

where $\mu_f$ and $\sigma_f$ are the mean and deviation of the objective function $f(x)$, respectively. Their values can be obtained through Monte Carlo simulation or the first order Taylor expansion if the design deviation of $x_i$ is small. When using Taylor expansions, $\mu_f$ and $\sigma_f$ can be represented by the following equations:

$$\begin{aligned} \mu_f &= f(x), \\ \sigma_f^2 &= \sum_{i=1}^{n} \left( \frac{\partial_f}{\partial_{x_i}} \right)^2 \partial_{x_i}^2, \end{aligned} \tag{3}$$

where $\sigma_{x_i}$ is the standard deviation of the $i$th $x$ component.

*3.2. Sensitivity Analysis.* There are 12 key points, and each one of them has 3 coordinate values. So, there are 36 coordinate parameters. If every one of the coordinate is selected as design variables, it needs much iteration. In order to reduce time of analysis and save resources, the full-factor test is utilized to determine main design variable, and the impact of every dependent variable is in Table 2. There are three levels, 1–3, and the larger the value, the greater the impact of the dependent variable. As shown in Table 2, Lca front $x$, Lca outer $x$, Uca front $x$, Uca front $y$, Uca back $x$, and Uca outer $x$ have made a minimal impact on sideways

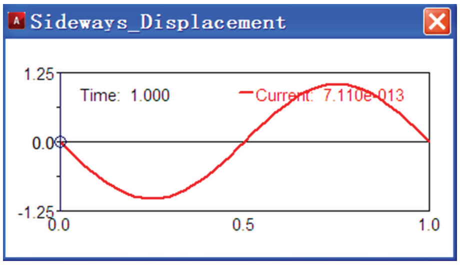


Figure 3: Sideways displacement.

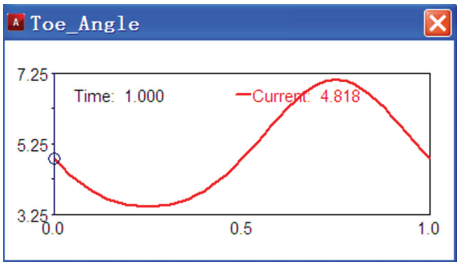


Figure 4: Toe angle.

Table 2: Impact of each variable.

| Coordinates of key points | Impact | |
|---|---|---|
| | Toe angle | Sideways displacement |
| Lca front $x$ | 1 | 1 |
| Lca front $y$ | 2 | 2 |
| Lca front $z$ | 3 | 3 |
| Lca back $x$ | 1 | 2 |
| Lca back $y$ | 1 | 3 |
| Lca back $z$ | 2 | 3 |
| Lca outer $x$ | 1 | 1 |
| Lca outer $y$ | 1 | 3 |
| Lca outer $z$ | 3 | 2 |
| Uca front $x$ | 1 | 1 |
| Uca front $y$ | 1 | 1 |
| Uca front $z$ | 3 | 3 |
| Uca back $x$ | 1 | 1 |
| Uca back $y$ | 1 | 2 |
| Uca back $z$ | 2 | 3 |
| Uca outer $x$ | 1 | 1 |
| Uca outer $y$ | 1 | 3 |
| Uca outer $z$ | 3 | 3 |

displacement and toe angle. Other coordinates of key points have made a great impact on sideways displacement and toe angle. Based on the test results, 12 main design variables are selected as controllable factors, and the variable name and its corresponding physical quantities are shown in Table 3.

*3.3. Kriging Model.* Engineering optimization problems often need enormous computation time for several programs running at the same time. We cannot provide the evaluation of the objective function and constraints to execute such large scale of exact analysis. So the application of approximation is necessary. In this paper, the Kriging model is adopted to build the approximation. Kriging model, one of the response surface models (RSM), has such advantages as unbiased estimator at the training sample point, desirably strong nonlinear approximating ability, and flexible parameter selection of the model, and thus it is quite suitable for approximate models [12]. Kriging models have a great promise for building accurate global approximations of a design space. These models are extremely flexible because of the wide range of spatial correlation functions that can be chosen for building the approximation, provided that sufficient sample data are available to capture the trends in the system responses; as a result, Kriging models can approach linear and nonlinear functions equally well. In addition, Kriging models can either "honor the data," by providing an exact interpolation of the data, or "smooth the data," by providing an inexact interpolation. One of the defects of using RSM in optimization is that it is apt to miss the global optimum because estimation value obtained with RSM includes errors at an unknown point [13].

Table 3: Controllable factors.

| Key point | Level 1 | Level 2 | Level 3 |
|---|---|---|---|
| Lca front $y(x_1)$ | −405 | −400 | −395 |
| Lca front $z(x_2)$ | 145 | 150 | 155 |
| Lca back $x(x_3)$ | 195 | 200 | 205 |
| Lca back $y(x_4)$ | −455 | −450 | −445 |
| Lca back $z(x_5)$ | 150 | 155 | 160 |
| Lca outer $y(x_6)$ | −755 | −750 | −745 |
| Lca outer $z(x_7)$ | 95 | 100 | 105 |
| Uca front $z(x_8)$ | 520 | 525 | 530 |
| Uca back $y(x_9)$ | −495 | −490 | −485 |
| Uca back $z(x_{10})$ | 525 | 530 | 535 |
| Uca outer $y(x_{11})$ | −680 | −675 | −670 |
| Uca outer $z(x_{12})$ | 520 | 525 | 530 |

In this paper the Kriging model is introduced into the robust design. In the conventional Kriging model, the performance $y(x)$ is modelled as follows:

$$y(x) = \beta^T h(x) + Z(x), \tag{4}$$

where $\beta^T h(x)$ is the regression component (e.g., a polynomial) which captures global trends; $Z(x)$ is assumed to be a Gaussian process indexed by input variables $x$, with zero mean and stationary covariance.

From a Bayesian perspective, the prior knowledge of the performance $y(x)$ is specified by a Gaussian process, which is characterized by the prior mean (i.e., the global trend) and prior covariance. Given the observations, the posterior process is also a Gaussian process (treating the covariance parameters as known and assuming a Gaussian prior distribution for $\beta$). The prediction of $y(x)$ is usually taken to be the posterior mean, and the prediction uncertainty is quantified by the posterior covariance.

The conventional Kriging model assumes that the Gaussian process has a stationary covariance, with the covariance function defined as follows:

$$C_{\mathrm{st}}(x_m, x_n; \Theta) = \sigma^2 \rho_{\mathrm{st}}(x_m, x_n; \theta), \tag{5}$$

where $\rho_{\mathrm{st}}$ is the correlation function. The hyper parameter set $\Theta$ is composed of $\{\sigma^2; \theta\}$. A frequently used Gaussian correlation function is

$$\rho_{\mathrm{st}}(x_m, x_n; \theta) = \exp\left[-\sum_{L=1}^{L} \theta^{(l)} \left(x_m^{(l)} - x_n^{(l)}\right)^2\right]. \tag{6}$$

The variance $\sigma^2$ provides the overall vertical scale relative to the mean of Gaussian process in the output space; $\theta = \{\theta^{(l)}\ (l = 1, 2, \ldots, L)\}$ are the correlation parameters (scaling factors) associated with each input variable $x^{(l)}$, which reflects the smoothness of the true performance. The stationary covariance indicates that the correlation function $\rho_{\mathrm{st}}(x_m, x_n; \theta)$ between any two sites $x_m$ and $x_n$ depends on only the distance (scaled by $\theta$) between $x_m$ and $x_n$ in (5) and (6); the subscript "st" means "stationary."

In order to innovate or improve and develop a new product and confirm a new technical parameter experiments usually need to be done repeatedly in the process of production and scientific research. It is very important to reasonably arrange experimental procedures to reduce the times of experiments and shorten the time of each experiment and avoid blindness. It requires two aspects of works to be done in order to solve the problem mentioned above. One is to design an experiment that can fully reflect the effect of all factors, which can reduce time of experiments and save resources. Another is to analyze the experimental results, in order to acquire reasonable conclusions and the error analysis.

DOE can analyze a design space and provide a rough estimate of an optimal design, which can be used as a starting point for numerical optimization. The Latin hypercube design could cover the design space more evenly than other DOE methods and generate more evenly distributed points. Therefore, in this paper, the Latin hypercube design is adopted to find the initial point and created the database for approximation model. For this problem, the inputs are the 12 main design variables, the outputs are the mean and their variance of the toe angle and sideways displacement, and 200 sample points from an LHS design are used to fit the kriging model. A set of 393 verification points, randomly selected across the domain, is used to evaluate the RMSE for each kriging model. Kriging model is established to fit the multiobjective robust design. The r-square of the kriging model is 0.87, so it can fit the virtual model.

## 4. Robust Design Based on Particle Swarm Optimization

The particle swarm optimization (PSO) is one of the evolutionary computation techniques introduced by Kennedy and Eberhart in 1995 [14]. It is a population-based search algorithm and is initialized with a population of random solutions, named particles; PSO makes use of a velocity vector to update the current position of each particle in the swarm [15, 16].

Particle swarm optimization is usually used as a traditional optimization method which is inspired from the social behaviour of flocks of birds. It is more competitive in various aspects, for example, due to its simplicity. Particle swarm optimization, genetic algorithms, and other evolutionary algorithms are all artificial life calculated. But particle swarm optimization is different from other evolutionary algorithms, using group iterative solution of cooperation mechanisms to generate the optimal solution instead of using group iterative solution of competing mechanisms. In PSO algorithm, each individual is called "particle," which represents a potential solution. The algorithm achieves the best solution by the variability of some particles in the tracing space. The particles search in the solution space following the best particle by changing their positions and the fitness frequently; the flying direction and velocity are determined by the objective function.

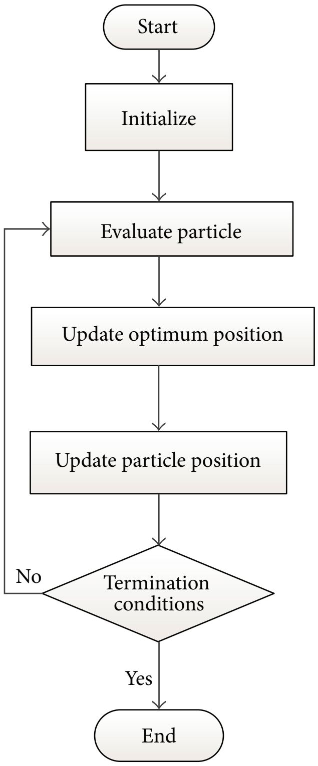


Figure 5: Particle swarm optimization.

The procedure of PSO is as follows:

(1) initialize the original position and velocity of particle swarm;

(2) calculate the fitness value of each particle;

(3) for each particle, compare the fitness value with the fitness value of pbest; if current value is better, then renew the position with current position, and update the fitness value simultaneously;

(4) determine the best particle of group with the best fitness value; if the fitness value is better than the fitness value of gbest, then update the gbest and its fitness value with the position;

(5) check the finalizing criterion; if it has been satisfied, quit the iteration; otherwise, return to step (2).

It can be shown as Figure 5.

Assuming $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ is the position of $i$th particle in D-dimension, $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ is its velocity which represents its direction of searching. In iteration process, each particle keeps the best position pbest found by itself; besides, it also knows the best position gbest searched by the group particles and changes its velocity according to the two best positions. The PSO is described in vector notation as to the follows:

$$\begin{aligned} v_i(t+l) &= \omega v_i(t) + c_1 r_1(t)\left(p_i(t) - x_i(t)\right) \\ &\quad + c_2 r_2(t)\left(pg(t) - x_i(t)\right), \quad i = 1, 2, \ldots, s, \quad (7) \\ x_i(t+1) &= x_i(t) + j + v_i(t+1), \end{aligned}$$

where $s$ is the swarm size. $c_1$ and $c_2$ are the nonnegative acceleration coefficients; these two constants make the particles have the ability of self-summary and learn from the excellent individuals of the groups, so the particles can close to the personal best solution of its own history and the global best solution within population or field. Typically value of $c_1$ and $c_2$ is 2. $\omega$ is the inertia weight, $r_l(t)$ and $r_2(t) \sim U(0,1)$, $x_i(t)$ is the position of particle $i$ at time $t$, $v_i(t)$ is the velocity of particle $i$ at time $t$, $p_i(t)$ is the personal best solution of particle $i$ at time $t$, and $pg(t)$ is the global best solution at time $t$.

The first term of (8) is the previous velocity of the particle vector. The second and third terms are used to change the velocity of the particle. Without the second and third terms, the particle will keep on "flying" in the same direction until it hits the boundary. The particle position $x(t+l)$ is updated using its current value and the newly computed velocity $v_i(t+l)$, which is determined by the values of $v_i(t)$, $x_i(t)$, $p_i(t)$, and $pg(t)$ and coefficients $\omega$, $c_1$, and $c_2$ [17].

In experiment, the population of group particle is 40; $c_1$, and $c_2$ are set to 2; the maximum time of iteration is 10000. It is acceptable if the difference between the best solution obtained by the optimization algorithm and the true solution is less than $1e-6$. The inertia weight is linear decreasing inertia all, which is determined by the following equation:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{\text{iter}_{\max}} \times k, \qquad (8)$$

where $w_{\max}$ is the start of inertia weight which is set to 0.9 and $w_{\min}$ is the end of inertia weight which is set to 0.05; $\text{iter}_{\max}$ is the maximum times of iteration; $k$ is the current iteration times. In order to reflect the universality of experiment, the original position and velocity are randomly generated.

Particle swarm optimization was used to search the optimal solution. A particle swarm optimization is created, the maximum iterations are set to 50, the number of particles is set to 15, and the objectives are the values and their variations of the toe angle and sideways displacement. V. Pareto, the French economist, who studied the multi-objective optimization problem of economics first, proposed the concept of Pareto solution set. There are 27 Pareto solutions in the results of the optimization.

In multiobjective optimization, each optimization objective is often conflicting, which requires coordination between the optimal solutions of each target. Considering the importance of each target, choose one Pareto optimal solution, and the design values of it are shown in Table 4. Using the results of the robust design to have a test in the ADAMS, the simulation results are shown in Figures 6 and 7.

It can be seen that the maximum deviation in the toe angle for the optimal design has been reduced by 52 percent, compared with the base design. As the discussion of the results, the most concerned factor is the relationship between objective function and design parameter. By comparing the experimental results, the robust design based on particle swarm significantly improved the robust of the toe angle and the sideways displacement, ensuring the reasonable of the design performance.

Table 4: Robust results.

| Key point | Initial value | Robust results |
|---|---|---|
| Lca front $y(x_1)$ | −400 | −400 |
| Lca front $z(x_2)$ | 150 | 150.33 |
| Lca back $x(x_3)$ | 200 | 205 |
| Lca back $y(x_4)$ | −450 | −450 |
| Lca back $z(x_5)$ | 155 | 151.67 |
| Lca outer $y(x_6)$ | −750 | −752.5 |
| Lca outer $z(x_7)$ | 100 | 101.33 |
| Uca front $z(x_8)$ | 525 | 519.67 |
| Uca back $y(x_9)$ | −490 | −489.67 |
| Uca back $z(x_{10})$ | 530 | 532.17 |
| Uca outer $y(x_{11})$ | −675 | −679.83 |
| Uca outer $z(x_{12})$ | 525 | 523.33 |

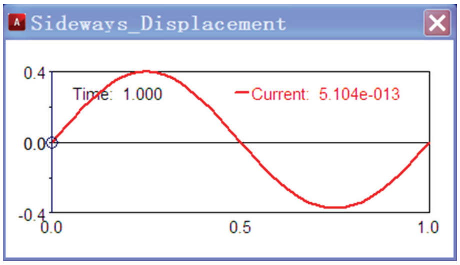


Figure 6: Sideways displacement.

## 5. Conclusion

In this study, a robust design based on bioinspired computation is presented and illustrated by the design of a double wishbone suspension system in order to reduce the effect of variations due to uncertainties in fabrication. As they are directly related to fabrication errors, the coordinates of key points were taken as design variables and at the same time are considered as random variables. So the robust design optimization problem had 13 design variables (joint positions) and 13 random constants (fabrication errors of joint positions). In this paper, the Latin hypercube design is adopted to make DOE design matrix of the 13 design variables. The Kriging model is built according to the result of DOE, and then the particle swarm is used to search optimal solution of the robust design. Particle swarm is implemented in a test case and the results show that the method can decrease the solution's time. The robustness of solution is improved. The improvement in robustness became larger as the amount of fabrication errors increases.

## Conflict of Interests

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

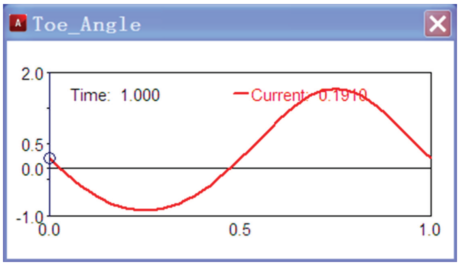


Figure 7: Toe angle.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under the Grant no. 51165007 and by the Natural Science Foundation of Jiangxi Province under the Grant no. 20132BAB206025. The authors would like to thank the reviewers for their valuable comments and suggestions.

## References

[1] J. C. Dixon, *Tires, Suspension and Handlin*, Society of Automotive Engineers, 2nd edition, 1996.

[2] G. Massimiliano and L. Francesco, “Multi-objective robust design of the suspension system of road vehicles,” *Vehicle System Dynamics*, vol. 41, pp. 537–546, 2004.

[3] F.-C. Wu and C.-C. Chyu, “Optimization of robust design for multiple quality characteristics,” *International Journal of Production Research*, vol. 42, no. 2, pp. 337–354, 2004.

[4] W.-J. Yin, Y. Han, and S.-P. Yang, “Dynamics analysis of air spring suspension system under forced vibration,” *China Journal of Highway and Transport*, vol. 19, no. 3, pp. 117–121, 2006.

[5] P.-M. Lu, L.-M. He, and J.-M. You, “Optimization of vehicle suspension parameters based on comfort and tire dynamic load,” *China Journal of Highway and Transport*, vol. 20, no. 1, pp. 112–117, 2007.

[6] H. Y. Kang and C. H. Suh, “Synthesis and analysis of spherical-cylindrical (SC) link in the McPherson strut suspension mechanism,” *Journal of Mechanical Design*, vol. 116, no. 2, pp. 599–606, 1994.

[7] T. Wang, “Multi-objective and multi-criteria decision optimization of automobile suspension parameters,” *Transactions of the Chinese Society of Agricultural Machinery*, vol. 28, no. 11, pp. 27–32, 2009.

[8] H. H. Chun, S. J. Kwon, and T. Tak, “Reliabilitybased design optimization of automotive suspension systems,” *International Journal of Automotive Technology*, vol. 8, no. 6, pp. 713–722, 2007.

[9] B.-L. Choi, J.-H. Choi, and D.-H. Choi, “Reliability-based design optimization of an automotive suspension system for enhancing kinematic and compliance characteristics,” *International Journal of Automotive Technology*, vol. 6, no. 3, pp. 235–242, 2004.

[10] S. R. Singiresu and K. A. Kiran, “Particle swarm methodologies for engineering design optimization,” in *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (ASME ’09)*, pp. 507–516, San Diego, Calif, USA, August 2009.

[11] J.-J. Chen, R. Xiao, Y. Zhong, and G. Dou, “Multidisciplinary robust optimization design,” *Chinese Journal of Mechanical Engineering*, vol. 18, no. 1, pp. 46–50, 2005.

[12] A. Giunta and L. T. Watson, *A Comparison of Approximation Modeling Technique: Polynomial Versus Interpolating Models*, AIAA/USAF/ NASA/ISSMO, 7th edition, 1998.

[13] R. J. Donald, S. Matthias, and J. W. William, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[14] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in *Proceedings of the IEEE International Conference on Neural Network*, pp. 1942–1948, Perth, Australia, December 1995.

[15] S. Mandal, R. Kar, D. Mandal, and S. P. Ghoshal, “Swarm intelligence based optimal linear phaseFIR high pass filter design using particle swarm optimization with constriction factor and inertia weight approach,” *World Academy of Science, Engineering and Technology*, vol. 5, no. 8, pp. 1155–1161, 2011.

[16] W.-M. Zhongi and S.-J. Li, “Feng QIAN. $\theta$-PSO: a new strategy of particle swarm optimization,” *Journal of Zhejiang University*, vol. 9, no. 6, pp. 786–790, 2008.

[17] S. Mandal and S. P. Ghshal, “Swarm intelligence based optimal linear fir high pass filter design using particle swarm optimization with constriction factor and inertia weight approach,” in *Proceedings of the IEEE Student Conference on Research and Development (SCOReD ’11)*, pp. 352–357, Cyberjaya, Malaysia, December 2011.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 731081, 7 pages
http://dx.doi.org/10.1155/2014/731081

*Research Article*

# Scheduling Projects with Multiskill Learning Effect

**Hong Zha and Lianying Zhang**

*College of Management and Economics, Tianjin University, No. 92 Weijin Road, Nankai District, Tianjin 300072, China*

Correspondence should be addressed to Lianying Zhang; tjzly126@126.com

Received 3 November 2013; Accepted 24 December 2013; Published 10 February 2014

Academic Editors: Q. Cheng and J. Yang

We investigate the project scheduling problem with multiskill learning effect. A new model is proposed to deal with the problem, where both autonomous and induced learning are considered. In order to obtain the optimal solution, a genetic algorithm with specific encoding and decoding schemes is introduced. A numerical example is used to illustrate the proposed model. The computational results show that the learning effect cannot be neglected in project scheduling. By means of determining the level of induced learning, the project manager can balance the project makespan with total cost.

## 1. Introduction

Project scheduling problem is a well-known research field in project management [1], while project scheduling with multiskilled resources is an important extension of it. In the extended problem, activities in the projects are assumed to require multiskills simultaneously to be executed. On the other hand, the resources involved (e.g., human resources) are assumed to possess multiskills and perform one of them at a time. As the assumption of it is closer to the real-world situation, the project scheduling problem with multiskills has attracted much attention in recent years [2–5].

Learning effect might be another nonignorable factor in actual scheduling [6, 7], due to the enormous effect on efficiency of operations. Anzanello and Fogliatto [8] were the pioneer introducing the learning effect into the machine scheduling problem. In addition, Biskup [7] classified the learning effect into two different approaches: the position-based learning and sum-of-processing-time based learning. The position-based learning was considered to be a realistic assumption for machine learning, while the sum-of-processing-time was regarded as a suitable hypothesis for human learning. Most of the researches were carried out based on above-mentioned assumptions. Cheng and Wang [9] simplified linear-log model into linear one in machine scheduling with position-based learning effect. Biskup and Simons [10] proposed the concept of induced learning and investigated to find out an optimal level of induced learning in machine scheduling. Wang [11] considered the machine scheduling problem affected by position-based learning as well as deteriorating jobs. Kuo and Yang [12] developed the model for single processor (e.g., human worker) scheduling problem with sum-of-processing-time based learning. Rudek [13] analyzed the single processor total weighted completion time scheduling problem with time-dependent learning effect and constructed a heuristic algorithm to solve it. Moreover, a few researchers discussed the scheduling problem where the two learning approaches coexisted in some situation. Wu and Lee [14] as well as Lu et al. [15] developed different models for the scheduling problem with both learning approaches.

Although the machine (processor) scheduling problems with learning effect have been fully investigated, the studies on project scheduling problems with learning effect have just started. Wu and Sun [16] were the first to develop a model for project scheduling and human resources allocation, but the situation where human resources possessed multiskills was not considered in the model. Janiak and Rudek [17] proposed a single processor (human worker) scheduling model with multiskill learning effect. But in the model the experience formula was oversimplified where the experience of a worker was assumed to be either 0 or 1; besides it could not be applied to project scheduling problem directly. Heimerl and Kolisch [18] formulated a work assignment model for multiskilled human resources, and many affecting factors were considered in the learning equation. But the equation took the form of position-based approach, which was unsuitable for human learning. Furthermore, the induced learning effect was not considered in the above-mentioned three papers. Their models could not assist the manager to determine an optimal level of induced learning.

To fill this gap, we propose a new model for project scheduling with multiskilled human resources, in which the learning formula is improved. In order to solve the proposed model, a genetic algorithm is introduced.

The remainder of the paper is organized as follows. Section 2 formulates the project scheduling problem considering multiskill learning effects. Genetic algorithm for solving this problem is described in Section 3. Numerical example is illustrated and discussed in Section 4. Finally, Section 5 concludes the paper.

## 2. Problem Formulation

We consider a project including $J$ activities. There are precedence relations between them due to technological requirements. The structure of the project can be depicted as a network $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. The nodes in the network represent activities, and the arcs between them represent precedence relations. Activity 1 is the source node of the network and activity $J$ is the sink node. They are both dummy activities. Each activity $j$ $(j = 2, \ldots, J-1)$ is performed by multiskilled human workers. Each worker assigned to the activity can only select one of the skills to perform. When the required quantities of each skill are available, the activity can be started. Once it is started, it may take a certain duration and not be preempted.

The efficiency of human workers might be influenced by learning effects during the project implement. Therefore, the actual duration for each activity might be adjusted in project scheduling. There are two kinds of learning, which are autonomous learning and induced learning. The autonomous learning is learning-by-doing, which means higher efficiency can be achieved by repeating similar activities. The induced learning is the proactive investment in technological knowledge, including worker's training, job-related instructions, and handbooks or supervisors. It can facilitate the learning effect so as to shorten the durations. However, extra cost may be also incurred [10]. Hence, the total cost for project may change too.

In this paper, we develop a novel model for the project scheduling problem with multiskilled worker, in which autonomous and induced learning are both considered. It can assist the project manager to determine the start time, the finish time, and resource allocation for each activity. Moreover, combined with the total cost formula, the project manager can balance the project makespan with the total cost. The proposed model is formulated as follows:

$$\min \quad T = f_J \tag{1}$$

$$\text{s.t}: \quad f_j - f_i \geq d_j, \quad i \in P_j \tag{2}$$

$$d_j = \max_{k=1,\ldots,K} \left\{ \sum_{l=1}^{L} \sum_{r=r_{jkls}}^{r_{jklf}} d_{lr} \right\} \tag{3}$$

$$d_{lr} = \begin{cases} d_{l1}, & r = 1 \\ \max \left\{ \left( \sum_{n=1}^{r-1} d_{ln} \right)^{\alpha_l}, d_{lm} \right\}, & r \geq 2 \end{cases} \tag{4}$$

$$a_l = \log_2 (1 - x_l) \, \mathrm{LR}_l, \quad 0 \leq x_l \leq x_{lm} < 1 \tag{5}$$

$$\sum_{j \in A_t} as_{jl} \leq \sum_{k=1}^{K} rs_{kl}, \quad A_t = \left\{ j \mid f_j - d_j < t \leq f_j \right\} \tag{6}$$

$$j = 1, \ldots, J, \quad k = 1, \ldots, K, \quad l = 1, \ldots, L. \tag{7}$$

In this model, the objective function (1) minimizes the project makespan $T$, where $f_J$ is the finish time of dummy activity $J$. Constraint (2) represents the precedence relations, where $P_j$ denotes the predecessors of activity $j$. The duration $d_j$ in constraint (2) is calculated by formulas (3)–(5).

According to formula (3), $d_j$ depends on the maximum time that all the workers spend in activity $j$. In formula (3), $d_{lr}$ denotes the time required for a worker to finish the $r$th unit of job $l$. The ordinal number $r_{jkls}$ means the accumulative units of job $l$ that worker $k$ have done since he finished the first unit of that job in activity $j$, while $r_{jkls}$ means the corresponding accumulative units when he finished the last unit of job $l$ in activity $j$.

Formula (4) is the sum-of-processing-time based learning function, which is proposed by Kuo and Yang [19]. In formula (4), $d_{l1}$ denotes the time required for a worker to finish the first unit of job using skill $l$, which is also the normal time for a worker to finish a unit of that job without learning effect. Parameter $d_{lm}$ is defined as the minimum value of $d_{lr}$. In addition, $a_l$ denotes the learning index of skill $l$, which reflects the learning speed on skill $l$.

The value of learning index can be calculated according to formula (5), which is proposed by Biskup and Simons [10]. In formula (5), $x_l$ denotes the induced learning level of skill $l$, and $x_{lm}$ is defined as the maximum value of $x_l$. Parameter $\mathrm{LR}_l$ indicates the autonomous learning ratio of skill $l$. When $x_l$ is equal to 0, it means that there is only autonomous learning effect existing for skill $l$. Otherwise, both autonomous and induced learning effects for skill $l$ are considered.

Constraint (6) represents that the quantity of skills required by activities cannot exceed what the workers can provide. Parameter $as_{jl}$ denotes the quantity of skill $l$ required by activity $j$, while 0-1 binary variable $rs_{jl}$ indicates whether worker $k$ possesses skill $l$ or not. When $rs_{jl}$ is equal to 1, worker $k$ possesses skill $l$ and vice versa.

Constraint (7) limits the range of $j$, $k$, and $l$.

The formula for extra cost incurred by induced learning is given below:

$$C = \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{l=1}^{L} c_l d_{jkl} + \sum_{l=1}^{L} b_{\,l} x_{\,l}^{2} + c_{\text{in}} T. \tag{8}$$

In formula (9), $c_l$ denotes the salary paid for a worker using skill $l$ per day, $b_l$ indicates the cost coefficient of induced learning for skill $l$, and $c_{\text{in}}$ is the daily indirect cost.

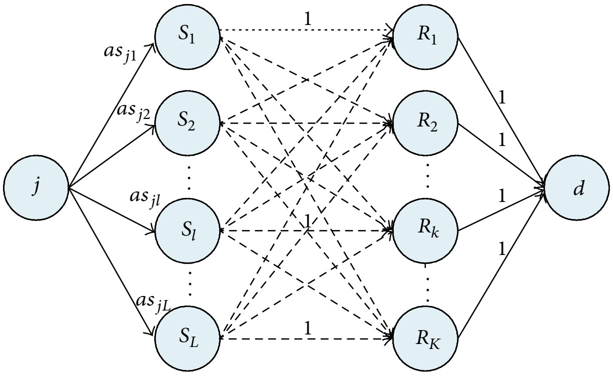


Figure 1: Maximum flow model for flexible resource allocation.

## 3. Genetic Algorithm

Genetic algorithm was proposed by Goldberg and Holland [20], which was inspired by Darwin's theory of evolution. According to the algorithm, the initial population with suitable encoding and decoding can evolve an optimal solution to real-world problem based on the natural principles of natural selection, crossover, and mutation. Due to its excellent performance, Genetic algorithm was widely used for project scheduling problem [21]. In this paper, we applied a genetic algorithm with specific coding and decoding mechanism to solve the above-mentioned formulation. The procedure of the genetic algorithm is described below.

*Step 1* (initial population). Genetic algorithm begins by means of generating an initial population randomly. Parameter pop is defined as the number of individuals in the population. In this paper, the individuals are encoded in the form of activity list. It is a permutation vector of activities conforming to precedence relation, which represents a feasible solution for schedule [22].

*Step 2* (fitness computation). To compute the fitness values, we decode the individuals with serial scheduling generation scheme [23], maximum flow algorithm [24], and EFT (early finish time) priority rule.

The serial scheduling generation scheme means the activities are selected according to their order in the list and scheduled at their earliest start period when the available resource is abundant. In order to determine whether the flexible resource allocation at one point is feasible or not, we introduce the maximum flow algorithm proposed by Ford and Fulkerson [24].

The flexible resource allocation can be modeled as a maximum flow network [25, 26], as shown in Figure 1. In the network, activity node $j$ is the source node, while dummy node $d$ is the sink node. Between source and sink node, there are skill nodes and resource nodes. The edges connecting activity node and skill nodes reflect the requirement of each skill for activity $j$, where the capacity for each edge represents the required quantity of each skill. While the edges linking resource and skill nodes illustrate the resource-skill relations. According to the relations, the capacity of each edge is set to be either 1 or 0. Besides, the edges connecting resource nodes and dummy nodes show the resource availability at a point in scheduling generation. When the resource is available, the capacity of the edge is set to be 1; otherwise, it is set to be 0. If the maximum flow obtained by the Ford and Fulkerson algorithm is equal to the sum of $as_{jl}$, the resource allocation for activity $j$ is feasible at this moment.

As there may be multiple feasible options for resource allocation of activity $j$, the eventual resource allocation can be determined by EFT priority rule. According to EFT rule, the option with minimized early finish time would be selected.

After decoding, the fitness for individual I can be calculated by the following:

$$\text{fitness}(i) = \frac{1}{T(i)}. \tag{9}$$

In formula (10), $T(i)$ denotes the makespan value of individual $i$.

*Step 3* (roulette wheel selection). In the selection stage, pop individuals are selected to reproduce according to their fitness values. With roulette wheel method, an individual can be chosen more than once. The selective probability of individual $i$ in population can be calculated by the following:

$$p(i) = \frac{\text{fitness}(i)}{\sum_{n=1}^{\text{pop}} \text{fitness}(n)}. \tag{10}$$

*Step 4* (partially matched crossover). In this step, the partially matched crossover operator is applied on two different individuals from current population with a probability of crossover $p_{\text{cos}}$.

For two parents $i^f$ and $i^m$ selected from the population, after an integer number $q_{\text{cos}}$ ($1 \leq q_{\text{cos}} < J$) is randomly generated, a son $i^s$ and $i^d$ are produced. For the son $i^s$, the first $q_{\text{cos}}$ activities are taken directly from $i^f$, while the remaining activities are arranged according to their order in $i^m$, while the daughter takes the first $q_{\text{cos}}$ activities from mother and rearranges the remaining according to father.

*Step 5* (swap mutation). In this step, the swap mutation operator is applied on newly generated individuals with a probability of mutation $p_{\text{mut}}$. Swap mutation can be described as follows.

A position $q_{\text{mut}}$ ($1 \leq q_{\text{mut}} \leq J$) is randomly chosen; then are swapped their contents if activity in position $q_{\text{mut}} + 1$ is not an immediate successor of activity in position $q_{\text{mut}}$. Otherwise, another position is randomly selected. This process repeats until the two adjacent activities can be permuted.

*Step 6* (stop criterion). The algorithm stops if the predefined evolutionary generation *gen* is reached. As the stop criterion is satisfied, the optimization result is given as output. Otherwise, the algorithm goes to Step 2 and continues to iterate.

## 4. Computational Experiment

In order to illustrate the model proposed in this paper, a numerical example is given below. As there are no benchmark

Table 1: Activities of example project.

| Activity no. | Successors | Normal duration (day) | Skill requirement | | | |
|---|---|---|---|---|---|---|
| | | | $\mathbf{s}_1$ | $\mathbf{s}_2$ | $\mathbf{s}_3$ | $\mathbf{s}_4$ |
| 1 | 2, 3, 4 | 0 | 0 | 0 | 0 | 0 |
| 2 | 11 | 22 | 0 | 2 | 0 | 2 |
| 3 | 5, 6, 7 | 25 | 2 | 0 | 0 | 3 |
| 4 | 9, 10, 11 | 9 | 1 | 0 | 3 | 0 |
| 5 | 8 | 19 | 0 | 1 | 4 | 0 |
| 6 | 9, 10 | 13 | 0 | 4 | 0 | 3 |
| 7 | 9 | 28 | 0 | 1 | 0 | 2 |
| 8 | 10 | 3 | 0 | 3 | 4 | 0 |
| 9 | 12 | 19 | 2 | 0 | 0 | 3 |
| 10 | 12 | 13 | 0 | 4 | 0 | 2 |
| 11 | 12 | 25 | 0 | 4 | 0 | 1 |
| 12 | — | 0 | 0 | 0 | 0 | 0 |

Table 2: Worker-skill mapping relationships.

| Worker | Skill | | | |
|---|---|---|---|---|
| | $\mathbf{s}_1$ | $\mathbf{s}_2$ | $\mathbf{s}_3$ | $\mathbf{s}_4$ |
| $\mathbf{w}_1$ | 1 | 1 | 1 | 1 |
| $\mathbf{w}_2$ | 1 | 0 | 1 | 1 |
| $\mathbf{w}_3$ | 1 | 1 | 1 | 0 |
| $\mathbf{w}_4$ | 0 | 1 | 1 | 1 |
| $\mathbf{w}_5$ | 1 | 1 | 0 | 1 |
| $\mathbf{w}_6$ | 1 | 0 | 1 | 0 |
| $\mathbf{w}_7$ | 0 | 1 | 1 | 0 |
| $\mathbf{w}_8$ | 0 | 1 | 0 | 1 |
| $\mathbf{w}_9$ | 1 | 0 | 0 | 1 |
| $\mathbf{w}_{10}$ | 0 | 0 | 1 | 1 |

Table 3: Learning parameters on skills.

| Skill | Autonomous learning ratio | Top level of induced learning | Minimum working time per unit of job (day) |
|---|---|---|---|
| $\mathbf{s}_1$ | 0.96 | 0.1 | 0.43 |
| $\mathbf{s}_2$ | 0.95 | 0.1 | 0.41 |
| $\mathbf{s}_3$ | 0.93 | 0.1 | 0.37 |
| $\mathbf{s}_4$ | 0.98 | 0.1 | 0.47 |

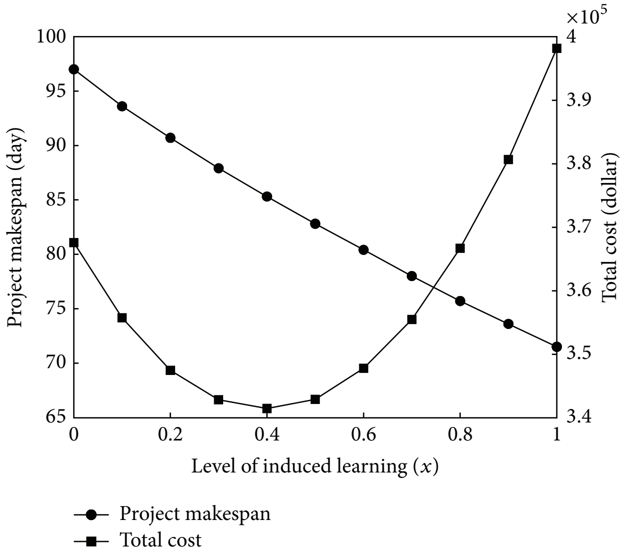


Figure 2: Project makespan-induced learning and total cost-induced learning curves.

examples for project scheduling problem with multiskill learning, we randomly generate an example project with multiskilled workers based on the existing research [2]. After that, the learning parameters and cost parameters are added into the generated example. In the example project, there are 12 activities, and the details of them are listed in Table 1. In addition, 10 workers ($\mathbf{w_1}$–$\mathbf{w_{10}}$) are involved in the project. Each worker is assumed to possess at least one of the four skills ($\mathbf{s_1}$–$\mathbf{s_4}$). The mapping relationships between the workers and the skills are present in Table 2. Besides, the related learning parameters for each skill are shown in Table 3. Finally, the cost parameters in the numerical are given below: $b_1 = 2.5 \times 10^6$ \$, $b_2 = 3.5 \times 10^6$ \$, $b_3 = 4.5 \times 10^6$ \$, $b_4 = 3.0 \times 10^6$ \$, $c_1 = 200$ \$, $c_2 = 400$ \$, $c_1 = 500$ \$, $c_3 = 300$ \$, and $c_{\text{in}} = 1200$ \$.

In the experiment, the parameters of genetic algorithm are set as follows: population size pop = 50, maximum generation gen = 100, crossover probability $p_{\cos} = 0.95$, and mutation probability $p_{\text{mut}} = 0.1$.

When the induced learning with top level is considered, the optimal solution is 1-3-2-5-8-6-4-7-10-11-9-12. After decoding it, we obtain a schedule with a makespan of 71.5 days. The optimal schedule is shown in Table 4. In addition, due to the learning effect, workers allocated to the same activity may have various working times. The details on working time in the aforementioned schedule are present in Table 5. From Tables 4 and 5, we can find out start and finish time for activities, resource allocation, and details on working time. For example, we learn that activity 7 is started on day 43.2 and finished on day 58.1. There are 3 workers ($\mathbf{w_1}$–$\mathbf{w_3}$) allocated to the activity. The workesr $\mathbf{w_1}$ and $\mathbf{w_2}$ are assigned to use skill $\mathbf{s_4}$, while worker $\mathbf{w_1}$ uses skill $\mathbf{s_1}$. In addition, all of them work in the activity for 14.9 days.

When no learning effect is considered in the project scheduling, the optimal solution is 1-2-4-3-11-7-5-6-8-9-10-12. The corresponding project makespan is 114 days and the optimal schedule is shown in Table 6.

Comparing the two schedules, we find that project makespan can be even shortened by 37.2% based on learning effect. The result demonstrated that learning effect cannot be neglected in project scheduling since it shortens the project makespan.

Furthermore, we define $x$ as the induced learning level of the four skills ($\mathbf{s_1}$–$\mathbf{s_4}$). Then, we discuss the changes in project makespan and total cost with $x$. The corresponding two curves are shown in Figure 2. It can be seen that the project makespan monotonically decreases when $x$ increases from 0 to 0.1. This is because the induced learning can help the workers improve their efficiency throughout the project.

Table 4: Optimal schedule with top level of induced learning.

| Activity no. | Start time (day) | Finish time (day) | Resource allocation | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\mathbf{w}_1$ | $\mathbf{w}_2$ | $\mathbf{w}_3$ | $\mathbf{w}_4$ | $\mathbf{w}_5$ | $\mathbf{w}_6$ | $\mathbf{w}_7$ | $\mathbf{w}_8$ | $\mathbf{w}_9$ | $\mathbf{w}_{10}$ |
| 1 | 0 | 0 | — | — | — | — | — | — | — | — | — | — |
| 2 | 0 | 15.8 | — | — | — | — | — | — | $\mathbf{s}_2$ | $\mathbf{s}_2$ | $\mathbf{s}_4$ | $\mathbf{s}_4$ |
| 3 | 0 | 17.6 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_1$ | $\mathbf{s}_4$ | $\mathbf{s}_1$ | — | — | — | — | — |
| 4 | 15.8 | 23 | — | — | — | — | — | $\mathbf{s}_3$ | $\mathbf{s}_3$ | | $\mathbf{s}_1$ | $\mathbf{s}_3$ |
| 5 | 17.6 | 30.7 | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_2$ | — | — | — | — | — |
| 6 | 33.6 | 43.2 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | — | $\mathbf{s}_2$ | $\mathbf{s}_2$ | — | — |
| 7 | 43.2 | 58.1 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | — | — | — | — | — | — | — |
| 8 | 30.7 | 33.6 | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_2$ | $\mathbf{s}_2$ | $\mathbf{s}_3$ | $\mathbf{s}_2$ | — | — | — |
| 9 | 58.1 | 71.5 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_1$ | — | — | $\mathbf{s}_1$ | — | — | — | $\mathbf{s}_4$ |
| 10 | 43.2 | 51.7 | — | — | — | $\mathbf{s}_2$ | $\mathbf{s}_2$ | — | $\mathbf{s}_2$ | $\mathbf{s}_2$ | $\mathbf{s}_4$ | $\mathbf{s}_4$ |
| 11 | 51.7 | 65.2 | — | — | — | $\mathbf{s}_2$ | $\mathbf{s}_2$ | — | $\mathbf{s}_2$ | $\mathbf{s}_2$ | $\mathbf{s}_4$ | — |
| 12 | 71.5 | 71.5 | — | — | — | — | — | — | — | — | — | — |

Table 5: Details on working time with top level of induced learning.

| Worker | Working time in each activity (day) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| $\mathbf{w}_1$ | 0 | 0 | 17.6 | 0 | 12.6 | 7.5 | 14.9 | 1.6 | 9.6 | 0 | 0 | 0 |
| $\mathbf{w}_2$ | 0 | 0 | 17.6 | 0 | 12.6 | 7.5 | 14.9 | 1.6 | 9.6 | 0 | 0 | 0 |
| $\mathbf{w}_3$ | 0 | 0 | 16.8 | 0 | 12.6 | 9.6 | 14.9 | 1.6 | 10 | 0 | 0 | 0 |
| $\mathbf{w}_4$ | 0 | 0 | 17.6 | 0 | 12.6 | 7.5 | 0 | 2.9 | 0 | 8.5 | 13.1 | 0 |
| $\mathbf{w}_5$ | 0 | 0 | 16.8 | 0 | 13.1 | 6.8 | 0 | 1.7 | 0 | 6.3 | 11.3 | 0 |
| $\mathbf{w}_6$ | 0 | 0 | 0 | 5.9 | 0 | 0 | 0 | 2.8 | 13.4 | 0 | 0 | 0 |
| $\mathbf{w}_7$ | 0 | 14.8 | 0 | 6.9 | 0 | 6.7 | 0 | 1.6 | 0 | 6.2 | 11.2 | 0 |
| $\mathbf{w}_8$ | 0 | 14.8 | 0 | 0 | 0 | 6.8 | 0 | 0 | 0 | 6.3 | 11.3 | 0 |
| $\mathbf{w}_9$ | 0 | 15.8 | 0 | 7.2 | 0 | 0 | 0 | 0 | 0 | 7.6 | 13.5 | 0 |
| $\mathbf{w}_{10}$ | 0 | 15.8 | 0 | 6.9 | 0 | 0 | 0 | 0 | 10.4 | 7.6 | 0 | 0 |

On the contrary, the total cost firstly decreases and then increases with increasing of $x$. This is because the induced learning can not only reduce cost by shortening the durations but also incur extra cost. When the reduced cost exceeds the incurred, the total cost decreases. Otherwise, the total cost increases.

The results illustrate that each level of induced learning corresponds to a combination of project makespan and total cost. By means of determining the level of induced learning, the project manager can balance them in project scheduling.

## 5. Conclusion

In this paper, we proposed a model for project scheduling problem with multiskill learning effect. In this model, both autonomous and induced learning are considered. Moreover, the learning function is sum-of-process time based, which is a suitable hypothesis for human learning. In order to solve the model, a genetic algorithm is introduced. The individuals in the algorithm are encoded in the form of activity list and decoded with SGS (serial generation scheme), maximum flow algorithm, and EFT (early finish time) priority rule. Partially matched crossover and swap mutation are applied to make the individuals feasible.

A numerical example is given to illustrate the model. The computational results show that learning effect cannot be neglected in project scheduling. The induced learning can affect both project makespan and total cost. By means of determining the level of induced learning, the project manager can balance them in project scheduling. Due to the uncertainty nature of project, the future research will extend the problem to uncertainty projects environment.

## Conflict of Interests

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work; there is no professional or other personal interests of any nature or kind in any product, service, and/or company that could be construed as influencing the position presented in, or the review of, the paper.

Table 6: Optimal schedule considering no learning effect.

| Activity no. | Start time (day) | Finish time (day) | Resource allocation | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\mathbf{w}_1$ | $\mathbf{w}_2$ | $\mathbf{w}_3$ | $\mathbf{w}_4$ | $\mathbf{w}_5$ | $\mathbf{w}_6$ | $\mathbf{w}_7$ | $\mathbf{w}_8$ | $\mathbf{w}_9$ | $\mathbf{w}_{10}$ |
| 1 | 0 | 0 | — | — | — | — | — | — | — | — | — | — |
| 2 | 0 | 22 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | $\mathbf{s}_2$ | — | — | — | — | — | — |
| 3 | 9 | 34 | — | — | — | — | $\mathbf{s}_4$ | $\mathbf{s}_1$ | — | $\mathbf{s}_4$ | $\mathbf{s}_1$ | $\mathbf{s}_4$ |
| 4 | 0 | 9 | — | — | — | — | $\mathbf{s}_1$ | $\mathbf{s}_3$ | $\mathbf{s}_3$ | — | — | $\mathbf{s}_3$ |
| 5 | 47 | 66 | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_2$ | — | $\mathbf{s}_3$ | — | — | — | — |
| 6 | 66 | 79 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | — | $\mathbf{s}_2$ | $\mathbf{s}_2$ | — | — |
| 7 | 34 | 62 | — | — | — | — | $\mathbf{s}_4$ | — | — | $\mathbf{s}_2$ | $\mathbf{s}_4$ | — |
| 8 | 79 | 82 | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_3$ | $\mathbf{s}_2$ | $\mathbf{s}_2$ | $\mathbf{s}_3$ | $\mathbf{s}_2$ | — | — | — |
| 9 | 82 | 101 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_1$ | $\mathbf{s}_4$ | $\mathbf{s}_1$ | — | — | — | — | — |
| 10 | 101 | 114 | $\mathbf{s}_4$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | $\mathbf{s}_2$ | $\mathbf{s}_2$ | — | $\mathbf{s}_2$ | — | — | — |
| 11 | 22 | 47 | $\mathbf{s}_2$ | $\mathbf{s}_4$ | $\mathbf{s}_2$ | $\mathbf{s}_2$ | — | — | $\mathbf{s}_2$ | — | — | — |
| 12 | 114 | 114 | — | — | — | — | — | — | — | — | — | — |

## Acknowledgment

This paper is supported by the National Natural Science Foundation of China (no. 71272146).

## References

[1] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 207, no. 1, pp. 1–14, 2010.

[2] E. Alba and J. Francisco Chicano, "Software project management with GAs," *Information Sciences*, vol. 177, no. 11, pp. 2380–2401, 2007.

[3] H. Li and K. Womer, "Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm," *Journal of Scheduling*, vol. 12, no. 3, pp. 281–298, 2009.

[4] C. Heimerl and R. Kolisch, "Scheduling and staffing multiple projects with a multi-skilled workforce," *OR Spectrum*, vol. 32, no. 2, pp. 343–368, 2010.

[5] V. Yannibelli and A. Amandi, "A knowledge-based evolutionary assistant to software development project scheduling," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8403–8413, 2011.

[6] M. J. Anzanello and F. S. Fogliatto, "Learning curve models and applications: literature review and research directions," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[7] D. Biskup, "A state-of-the-art review on scheduling with learning effects," *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[8] M. J. Anzanello and F. S. Fogliatto, "Learning curve models and applications: literature review and research directions," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[9] T. E. Cheng and G. Wang, "Single machine scheduling with learning effect considerations," *Annals of Operations Research*, vol. 98, no. 1–4, pp. 273–290, 2000.

[10] D. Biskup and D. Simons, "Common due date scheduling with autonomous and induced learning," *European Journal of Operational Research*, vol. 159, no. 3, pp. 606–616, 2004.

[11] J.-B. Wang, "A note on scheduling problems with learning effect and deteriorating jobs," *International Journal of Systems Science*, vol. 37, no. 12, pp. 827–833, 2006.

[12] W.-H. Kuo and D.-L. Yang, "Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect," *European Journal of Operational Research*, vol. 174, no. 2, pp. 1184–1190, 2006.

[13] R. Rudek, "The single processor total weighted completion time scheduling problem with the sum-of-processing-time based learning model," *Information Sciences*, vol. 199, pp. 216–229, 2012.

[14] C.-C. Wu and W.-C. Lee, "Single-machine and flowshop scheduling with a general learning effect model," *Computers and Industrial Engineering*, vol. 56, no. 4, pp. 1553–1558, 2009.

[15] Y.-Y. Lu, C.-M. Wei, and J.-B. Wang, "Several single-machine scheduling problems with general learning effects," *Applied Mathematical Modelling*, vol. 36, no. 11, pp. 5650–5656, 2012.

[16] M.-C. Wu and S.-H. Sun, "A project scheduling and staff assignment model considering learning effect," *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 11-12, pp. 1190–1195, 2006.

[17] A. Janiak and R. Rudek, "A note on a makespan minimization problem with a multi-ability learning effect," *Omega*, vol. 38, no. 3-4, pp. 213–217, 2010.

[18] C. Heimerl and R. Kolisch, "Work assignment to and qualification of multi-skilled human resources under knowledge depreciation and company skill level targets," *International Journal of Production Research*, vol. 48, no. 13, pp. 3759–3781, 2010.

[19] W.-H. Kuo and D.-L. Yang, "Minimizing the makespan in a single machine scheduling problem with a time-based learning effect," *Information Processing Letters*, vol. 97, no. 2, pp. 64–67, 2006.

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

[21] R. Kolisch and S. Hartmann, "Experimental investigation of heuristics for resource-constrained project scheduling: an update," *European Journal of Operational Research*, vol. 174, no. 1, pp. 23–37, 2006.

[22] K. Moumene and J. A. Ferland, "Activity list representation for a generalization of the resource-constrained project scheduling

problem," *European Journal of Operational Research*, vol. 199, no. 1, pp. 46–54, 2009.

[23] R. Kolisch, "Serial and parallel resource-constrained project scheduling methods revisited: theory and computation," *European Journal of Operational Research*, vol. 90, no. 2, pp. 320–333, 1996.

[24] L. Ford and D. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[25] J. Yang and Z. Fei, "Bipartite graph based dynamic spectrum allocation for wireless mesh networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 96–101, IEEE press, June 2008.

[26] C. Liu, G. Chang, and J. Jia, "A new channel assignment algorithm in wireless mesh network," in *Recent Advances in Computer Science and Information EngIneerIng*, vol. 127 of *Lecture Notes in Electrical Engineering*, pp. 511–516, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 215472, 16 pages
http://dx.doi.org/10.1155/2014/215472

*Research Article*

# An Adaptive Hybrid Algorithm Based on Particle Swarm Optimization and Differential Evolution for Global Optimization

**Xiaobing Yu,[1,2] Jie Cao,[1,2] Haiyan Shan,[1] Li Zhu,[1] and Jun Guo[3]**

[1] *China Institute of Manufacturing Development, Nanjing University of Information Science & Technology, Nanjing 210044, China*
[2] *Collaborative Innovation Center on Forecast and Evaluation of Meteorological Disasters, Nanjing University of Information Science & Technology, Nanjing 210044, China*
[3] *School of Mechanic and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China*

Correspondence should be addressed to Xiaobing Yu; yuxb111@163.com

Received 30 September 2013; Accepted 17 November 2013; Published 9 February 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

Particle swarm optimization (PSO) and differential evolution (DE) are both efficient and powerful population-based stochastic search techniques for solving optimization problems, which have been widely applied in many scientific and engineering fields. Unfortunately, both of them can easily fly into local optima and lack the ability of jumping out of local optima. A novel adaptive hybrid algorithm based on PSO and DE (HPSO-DE) is formulated by developing a balanced parameter between PSO and DE. Adaptive mutation is carried out on current population when the population clusters around local optima. The HPSO-DE enjoys the advantages of PSO and DE and maintains diversity of the population. Compared with PSO, DE, and their variants, the performance of HPSO-DE is competitive. The balanced parameter sensitivity is discussed in detail.

## 1. Introduction

Evolutionary algorithms (EAs), inspired by the natural evolution of species, have been successfully applied to solve numerous optimization problems in diverse fields [1]. Particle swarm optimization (PSO) and differential evolution (DE) are two stochastic, population-based optimization EAs [2].

PSO was introduced by Kennedy and Eberhart in 1995 [3, 4]. PSO uses a simple mechanism that mimics swarm behavior in birds flocking and fish schooling to guide the particles to search for globally optimal solutions. As PSO is easy to implement, it has rapidly progressed in recent years with many successful applications in solving real-world optimization problems. During the last decade, PSO algorithm has been paid much attention to in various areas [5–19] due to its effectiveness in handling optimization problems. Unfortunately, the PSO algorithm suffers from the premature convergence problem which does exist in complex optimization issues. In [17–19], some methods for tuning parameters including inertia weights and acceleration coefficients for PSO have been proposed to enhance PSO's search performance. A comprehensive learning PSO (CLPSO) algorithm was proposed in [14], which shows its superiority in dealing with multimodal functions.

DE is a simple yet powerful EA for global optimization introduced by Storn and Price [20]. The DE algorithm has gradually become more popular and has been used in many practical cases, mainly because it has demonstrated good convergence properties and is principally easy to understand [21]. DE has been successfully applied in diverse fields of engineering [22–26]. The performance of the conventional DE algorithm highly depends on the chosen trial vector generation strategy and associated parameter values used. Inappropriate choice of strategies and parameters may lead to premature convergence, which have been extensively demonstrated in [27]. In the past decade, DE researchers have suggested many empirical guidelines for choosing trial vector generation strategies and their associated control parameter settings [1, 28–31].

Although DE and PSO have been successfully applied to a wide range of problems including test and real life problems, both have certain shortcomings associated with them which sometimes deteriorate the performance of algorithms. The major problem is the lack of diversity resulting in a suboptimal solution or a slow convergence rate [32]. In order to improve the performance of these algorithms, one of the class of modified algorithms consists of the hybridization of algorithms, where the two algorithms are combined together to form a new algorithm. DE is applied to each particle for a finite number of iterations to determine the best particle which is then included into the population [33]. DE is applied to the best particle obtained by PSO [34]. A hybrid version of PSO and DE is proposed which is named Barebones DE [35]. The evolution candidate solution is generated either by DE or by PSO according to some fixed probability distribution [36]. A hybrid metaheuristic is designed so as to preserve the strengths of both algorithms [32].

However, it is worth mentioning that, in almost all the hybrid works mentioned above, the convergence rate is not fast enough or the global search performance is not satisfactory. To achieve the goal, a novel adaptive hybrid algorithm based on PSO and DE (HPSO-DE) is formulated by developing a balanced parameter between PSO and DE. Adaptive mutation is carried out to current population when the population clusters around local optima. The HPSO-DE enjoys the advantages of PSO and DE and maintains diversity of the population. With the efforts, the HPSO-DE has the ability to jump out of the local optima.

## 2. PSO and DE

*2.1. PSO.* In the standard PSO, a swarm consists of $m$ individuals (called particles) that fly around in an $n$-dimensional search space. The position of the $i$th particle at the $t$th iteration is used to evaluate the particle and represent the candidate solution for the optimization problem. It can be represented as $X_i^t = [x_{i1}^t, x_{i2}^t, \ldots, x_{in}^t]$, where $x_{ij}^t$ is the position value of the $i$th particle with respect to the $j$th dimension ($j = 1, 2, \ldots, n$). During the search process, the position of a particle is guided by two factors: the best position visited by itself ($p_{\text{best}}$) denoted by $P_i^t = [p_{i1}^t, p_{i2}^t, \ldots, p_{in}^t]$ and the position of the best particle found so far in the swarm ($g_{\text{best}}$) denoted by $G^t = [g_1^t, g_2^t, \ldots, g_n^t]$. The new velocity (denoted by $V_i^t = [v_{i1}^t, v_{i2}^t, \ldots, v_{in}^t]$) and position of particle $i$ at the next iteration are calculated according to

$$v_{ij}^{t+1} = w \times v_{ij}^t + c_1 \times r_1 \times \left(p_{ij}^t - x_{ij}^t\right) + c_2 \times r_2 \times \left(g_j^t - x_{ij}^t\right), \tag{1}$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1}, \tag{2}$$

where $w$ is the inertia weight, $c_1$, and $c_2$ are, respectively, the cognitive and social learning parameters, and $r_1$, and $r_2$ are random numbers between $(0, 1)$. Based on the above equations, the particle can fly through search space toward $p_{\text{best}}$ and $g_{\text{best}}$ in a navigated way [16, 17].

*2.2. PSO Variants*

*2.2.1. PSO-w.* In the PSO algorithm, proper control of global exploration and local exploitation is an important issue. In general, the higher values of inertia weight $w$ help in exploring the search space more thoroughly in the process and benefit the global search, while lower values help in the local search around the current search space. The major concern of this linear PSO is to avoid the premature convergence in the early period of the search and to enhance convergence to the global optimum solution during the latter period of the search. The concept of linearly decreasing inertia weight was introduced in [17] and is given by

$$w = w_{\max} - \left(w_{\max} - w_{\min}\right)\left(\frac{\text{iter}}{\text{iter}_{\max}}\right), \tag{3}$$

where iter is the current iteration number and $\text{iter}_{\max}$ is the maximum number of iteration. Usually the value of $w$ is between 0.9 and 0.4. Therefore, the particle is to use lager inertia weight during the initial exploration and gradually reduce its value as the search proceeds in further iterations. According to the research [37], the inertia weight is adjusted by (4). The nonlinear descending can achieve faster convergence speed than that with linear inertia weight:

$$w = \left(w_1 - w_2\right) \times \frac{\left(\text{iter} - \text{iter}_{\max}\right)^2}{\left(\text{iter}_{\max}\right)^2} + w_2, \tag{4}$$

where $w_1$ and $w_2$ are the initial and final inertia weight.

*2.2.2. PSO-TVAC.* Although linear PSO can locate satisfactory solution at a markedly fast speed, its ability to fine-tune the optimum solution is limited, mainly due to the lack of diversity at the latter stage of evolution process. In population-based optimization methods, the guideline is to encourage the individuals to roam through the entire search space during the early period of the search, without clustering around local optima. During the later period, convergence towards the global optima is encouraged [3]. With this view, a novel strategy in which time-varying acceleration coefficients are employed by changing the acceleration coefficients with time is proposed [16, 38]. With a large cognitive component and small social component at the beginning, particles are allowed to move around the search space, instead of moving toward the population best. On the other hand, a small cognitive component and a large social component allow the particles to converge to the global optima in the latter part of the optimization. This approach is referred to as PSO-TVAC. This modification can be mathematically represented as follows:

$$\begin{gathered} c_1 = \left(c_{1f} - c_{1i}\right)\left(\frac{\text{iter}}{\text{iter}_{\max}}\right) + c_{1i}, \\ c_2 = \left(c_{2f} - c_{2i}\right)\left(\frac{\text{iter}}{\text{iter}_{\max}}\right) + c_{2i}, \end{gathered} \tag{5}$$

where $c_{1i}$, $c_{1f}$, $c_{2i}$, and $c_{2f}$ are initial and final values of cognitive and social acceleration factors, respectively; usually $c_{1i} = c_{2f} = 2.5$ and $c_{1f} = c_{2i} = 0.5$ [38].

*2.3. DE.* DE is proposed by Storn and Price [20]. It is an effective, robust, and simple global optimization algorithm. According to frequently reported experimental studies, DE has shown better performance than many other EAs in terms of convergence speed and robustness over several benchmark functions and real-world problems [39].

In DE, there are three operators: mutation, crossover, and selection. Initially, a population is generated randomly with uniform distribution; then the mutation, crossover, and selection operators are applied to generate a new population. Trial vector generation is a crucial step in DE process. The two operators mutation and crossover are used to generate the trial vectors. The selection operator is used to select the best trial vector for the next generation. The initialization and DE operators are explained briefly as follows [40].

DE starts with a population of NP $D$-dimensional candidate solutions which may be represented as $X_{i,G}(i = 1,2,\ldots,\text{NP}) = \{x_{i,G}^1, x_{i,G}^2, \ldots, x_{i,G}^D\}$, where index $i$ denotes the $i$th individual of the population, $G$ denotes the generation to which the population belongs, and $D$ is the dimension of the population.

The initial population should try to cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the minimum $X_{\min} = \{x_{\min}^1, x_{\min}^2, \ldots, x_{\min}^D\}$ and maximum $X_{\max} = \{x_{\max}^1, x_{\max}^2, \ldots, x_{\max}^D\}$ bounds. Thus, the initial population can be described as follows:

$$x_{i,0} = x_{\min} + \text{rand}(0,1) \times (x_{\max} - x_{\min}), \tag{6}$$

where rand$(0,1) \in [0,1]$ is a uniformly distributed random variable [40, 44].

*(1) Mutation.* After initialization, DE utilizes the mutation operation to generate a trial vector $V_{i,G} = \{v_{i,G}^1, v_{i,G}^2, \ldots, v_{i,G}^D\}$ with respect to each individual in the current population. $V_{i,G}$ can be produced by certain mutation strategy. For example, the five most frequently mutation strategies implemented in the DE are listed as follows [1]:

DE/rand/1:

$$V_{i,G} = X_{r_1^i,G} + F \cdot \left(X_{r_2^i,G} - X_{r_3^i,G}\right), \tag{7}$$

DE/best/1:

$$V_{i,G} = X_{\text{best},G} + F \cdot \left(X_{r_1^i,G} - X_{r_2^i,G}\right), \tag{8}$$

DE/rand-to-best/1:

$$\begin{aligned} V_{i,G} = X_{i,G} &+ F \cdot (X_{\text{best},G} - X_{i,G}) \\ &+ F \cdot \left(X_{r_1^i,G} - X_{r_2^i,G}\right), \end{aligned} \tag{9}$$

DE/best/2:

$$\begin{aligned} V_{i,G} = X_{\text{best},G} &+ F \cdot \left(X_{r_1^i,G} - X_{r_2^i,G}\right) \\ &+ F \cdot \left(X_{r_3^i,G} - X_{r_4^i,G}\right), \end{aligned} \tag{10}$$

DE/rand/2:

$$V_{i,G} = X_{r_1^i,G} + F \cdot \left(X_{r_2^i,G} - X_{r_3^i,G}\right) + F \cdot \left(X_{r_4^i,G} - X_{r_5^i,G}\right). \tag{11}$$

The indices $r_1^i$, $r_2^i$, $r_3^i$, $r_4^i$, and $r_5^i$ are mutually exclusive integers randomly generated within the range $[0,1]$, which are also different from the index $i$ [1]. $F$ is the mutation scale factor which is used in controlling the amplification of the differential variation [40].

*(2) Crossover.* The crossover operation is introduced to increase the diversity of the target vectors. After the mutation phase, the crossover operation is applied to $V_{i,G} = \{v_{i,G}^1, v_{i,G}^2, \ldots, v_{i,G}^D\}$ and $X_{i,G} = \{X_{i,G}^1, X_{i,G}^2, \ldots, X_{i,G}^D\}$ to generate a trial vector $U_{i,G} = \{u_{i,G}^1, u_{i,G}^2, \ldots, u_{i,G}^D\}$ as follows:

$$u_{i,G}^j = \begin{cases} v_{i,G}^j & \text{if rand}_j\,[0,1) \le \text{CR or } \left(j = j_{\text{rand}}\right), \\ x_{i,G}^j & \text{others.} \end{cases} \tag{12}$$

CR $\in [0,1]$ is the crossover constant, which has to be determined by the user. $j_{\text{rand}} \in [1,D]$ is a randomly chosen index which ensures that the trial vector $U_{i,G}$ will differ from $X_{i,G}$ by at least one parameter.

*(3) Selection.* If the generated trial vector $u_{i,G}^j$ exceeds the corresponding upper and lower bounds, we randomly and uniformly reinitialize it within the search range. Then the fitness values of all trial vectors are evaluated [45].

After that, a greedy selection scheme is employed:

$$x_{i,G+1} = \begin{cases} u_{i,G} & \text{if } f\left(u_{i,G}\right) \le f\left(x_{i,G}\right), \\ x_{i,G} & \text{otherwise.} \end{cases} \tag{13}$$

If the trial vector $u_{i,G}$ yields a better cost function value than $x_{i,G}$, the $u_{i,G}$ will replace $x_{i,G}$ and enter the population of the next generation; otherwise, the old value $x_{i,G}$ is retained.

*2.4. DE Variants.* In order to improve the performance of DE, some adaptive DE variants are proposed. jDE was proposed based on the self-adaptation of the scale factor $F$ and the crossover rate CR [29]. The jDE is represented by a $D$-dimensional vector $X_{i,G}(i = 1,2,\ldots,\text{NP})$. New control parameters or factors $F_{i,G+1}$ and $\text{CR}_{i,G+1}$ are calculated as

$$\begin{aligned} F_{i,G+1} &= \begin{cases} F_l + \text{rand}_1 \times F_u & \text{if rand}_2 < \tau_1, \\ F_{i,G} & \text{otherwise,} \end{cases} \\ \text{CR}_{i,G+1} &= \begin{cases} \text{rand}_3 & \text{if rand}_4 < \tau_2, \\ \text{CR}_{i,G} & \text{otherwise,} \end{cases} \end{aligned} \tag{14}$$

and they produce factors $F$ and CR in a new parent vector. rand$_j$ ($j \in \{1,2,3,4\}$) are uniform random values within the range $[0,1]$. $\tau_1$ and $\tau_2$ represent probabilities to adjust factors $F$ and CR, respectively. Generally, $\tau_1 = \tau_2 = 0.1$, $F_l = 0.1$, $F_u = 0.9$. SaDE gave the first attempt to simultaneously adopt more than one mutation scheme in DE [1]. The main propose is to reduce the problem-solving risk by distributing available computational resources to multiple search techniques with different biases. SaNSDE can be regarded as an improved version of the SaDE. Its mutation is executed in the same way as SaDE except that only two mutation schemes are used, and the scale factor $F$ in the adopted mutation schemes is generated according to either a Gaussian distribution or a Cauchy distribution [30]. JADE is another recent DE variant, in which a new mutation scheme named "/DE/current-to-pbest" is adopted [31].

## 3. Proposed HPSO-DE

Similar to other EAs, both of PSO and DE are the population-based iterative algorithms. The PSO and DE can easily get trapped in the local optima when solving complex multimodal problems. These weaknesses have restricted wider applications of them. Therefore, avoiding the local optima has become one of the most important and appealing goals in algorithms research. To achieve the goal, an adaptive hybrid algorithm based on PSO and DE (HPSO-DE) for global optimization is proposed. The algorithm focuses on the convergence of population. When a particle discovers a current optima position, the other particles will draw together to the particle. If the position is the local optima, the algorithm will be convergence and clustered in local optima. The premature may appear. Suppose that the population size of HPSO-DE is NP, the fitness value of $i$th particle is $f_i$, and the average fitness value is $f_{\text{avg}}$. The convergence degree is defined as follows:

$$d = \sqrt{\sum_{i=1}^{NP}\left(\frac{f_i - f_{\text{avg}}}{\max\{1, \max_{1\le i\le N}(f_i - f_{\text{avg}})}\right)^2}. \qquad (15)$$

The parameter $d$ reflects the convergence degree. When the parameter $d$ is large, the algorithm is in random search. On the other hand, the algorithm will get into local optima and the premature maybe occur. In order to evaluate the parameter $d$, $d_c$ is given as follows, where $p$ is the mutation probability.

$$p = \begin{cases} k, & d < d_c, \\ 0, & \text{others}. \end{cases} \qquad (16)$$

Generally, $d_c \in [0.5, 2]$. If the parameter $d$ is less than $d_c$, the mutation probability $p$ is equal to $k$. The mutation including PSO and DE is as follows, respectively:

$$g_{\text{best}} = (1 + 0.5\eta)\, g_{\text{best}}, \qquad (17)$$

$$X_{i,G} = (1 + 0.5\eta)\, X_{i,G}, \qquad (18)$$

where the parameter $\eta$ obeys Gauss (0, 1) distribution.

The strategy makes HPSO-DE enjoy the advantages of two algorithms and maintain diversity of the population. With the efforts, the HPSO-DE has the ability of jumping out of the local optima. The main procedure of HPSO-DE is presented in Algorithm 1.

One of the most important parameters for the proposed HPSO-DE is the balanced parameter $p$. In the following, we will make an integrated analysis of the key parameter by comparing the performance of the HPSO-DE in the optimization of several representative functions.

## 4. Numerical Experiments and Results

*4.1. Test Functions.* 16 benchmark functions are used to test the performance of HPSO-DE to assure a fair comparison. If the number of test problems is smaller, it will be very difficult to make a general conclusion. Using a test set which is too small also has the potential risk that the algorithm is biased (optimized) toward the chosen set of problems. Such bias might not be useful for other problems of interest. The benchmark functions are given in Table 1. It denotes the ranges of the variables and the value of the global optimum. Functions $f_1 - f_{16}$ are high-dimensional problems. Functions $f_1 - f_6$ are unimodal. Function $f_7$ is a noisy quadratic function. Functions $f_8 - f_{16}$ are multimodal functions where the number of local minima increases exponentially with the problem dimension [29]. $f_{15} - f_{18}$ are rotated functions. An orthogonal matrix $M$ is generated to rotate a function. The original variable $x$ is left-multiplied by the orthogonal matrix $M$ to get the new rotated variable $y = M \times x$. This variable $y$ is used to compute the fitness value $f$. When one dimension in $x$ is changed, all dimensions in $y$ will be influenced:

$$f_1 = \sum_{i=1}^{D} x_i^2,$$

$$f_2 = \sum_{i=1}^{D} i x_i^2,$$

$$f_3 = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|,$$

$$f_4 = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2,$$

$$f_5 = \max_i \{|x_i|, 1 \le x_i \le D\},$$

$$f_6 = \sum_{i=1}^{D-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right],$$

$$f_7 = \sum_{i=1}^{D} i x_i^4 + \text{random}[0, 1),$$

$$f_8 = \sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right],$$

$$f_9(x) = \sum_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right),$$

$$y_i = \begin{cases} x_i, & |x_i| < 0.5, \\ \dfrac{\text{round}(2x_i)}{2}, & |x_i| \ge 0.5, \end{cases}$$

```
Step 1. Initialize parameters: PSO-Flag=false; DE-Flag=false; the mutation probability of PSO: $\mathrm{PSO}_p$;
        the mutation probability of DE: $\mathrm{DE}_p$; the balanced parameter between PSO and DE: $p$.
        Convergence evaluation parameter $d_c$.
Step 2. generation counter $G = 0$ and randomly initialize a population of NP individuals
        $P_G = \{X_{1,G}, \ldots, X_{\mathrm{NP},G}\}$ with $X_{i,G} = \{x_{i,G}^1, \ldots, x_{i,G}^D\}$, $i = 1, \ldots,$ NP uniformly distributed in the range
        $[X_{\min}, X_{\max}]$, where $X_{\min} = \{x_{\min}^1, x_{\min}^2, \ldots, x_{\min}^D\}$ and $X_{\max} = \{x_{\max}^1, x_{\max}^2, \ldots, x_{\max}^D\}$.
Step 3. Evaluation the population and Identify the best position.
Step 4. While stopping criterion is not satisfied
        Generate Random number rand in (0, 1).
        If rand < $p$
                    Set DE-Flag=true; %%Active jDE for current population
                    for $i = 1$ to the NP do
                        Update $F$ and $R$ according to (14);
                        Generate $V_{i,G}$ using (7)
                        Generate the trial vector $U_{i,G}$ by (12)
                End for
        Else
            Set PSO-Flag=true; %%Active PSO for current population
            Update $w$ using (4)
            for $i = 1$ to the NP do
                Update particle Velocity and Position according to (1), (2) respectively.
                Set the values of position to the trial vector $U_{i,G}$.
            End for
        End if
        Step 4.1. Randomly reinitialize the trial vector $U_{i,G}$ within the search
                  space if any variable is outside its boundaries.
        Step 4.2. Selection
             for $i = 1$ to NP
                        Evaluate the trial vector $U_{i,G}$
                              If $f(U_{i,G}) \le f(X_{i,G})$
                                 $X_{i,G+1} = U_{i,G}, f(X_{i,G+1}) = f(U_{i,G})$
                                 $P_{i,\mathrm{pbest}} = U_{i,G}, f(P_{i,\mathrm{pbest}}) = f(U_{i,G})$
                                     If $f(U_{i,G}) < f(X_{\mathrm{best},G})$
                                        $X_{\mathrm{best},G} = U_{i,G}, f(X_{\mathrm{best},G}) = f(U_{i,G})$
                                        $p_g = U_{i,G}, f(p_g) = f(U_{i,G})$
                                   End if
                             End if
            End for
        Step 4.3. Calculate parameter $d$ using (15)
             If parameter meets the requirement of (16)
                    Generate a random number rand in (0, 1).
                    If rand is less than $\mathrm{PSO}_p$ and PSO-Flag is true
                       Update $g_{\mathrm{best}}$ using (17)
                    End if
                    If rand is less than $\mathrm{DE}_p$ and DE-Flag is true
                          Update $X_{i,G+1}$ using (18)
                    End if
             End if
        Step 4.4. Increment the generation count $G = G + 1$;
            Reset PSO-Flag=false; DE-Flag=false;
Step 5. End while
```

Algorithm 1

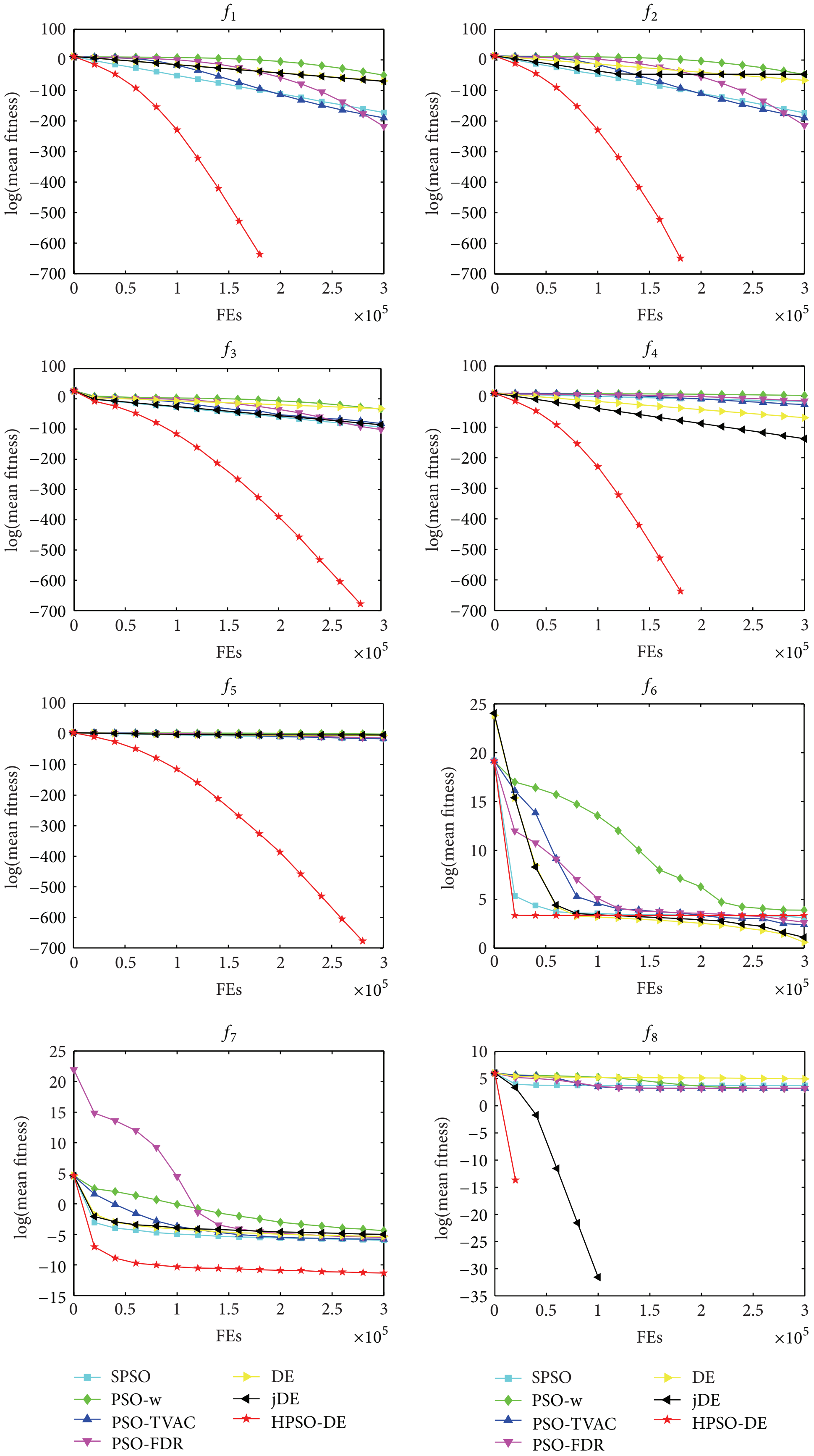


Figure 1: Performance of the algorithms for $f_1$, $f_2$, $f_3$, $f_4$, $f_5$, $f_6$, $f_7$, and $f_8$.

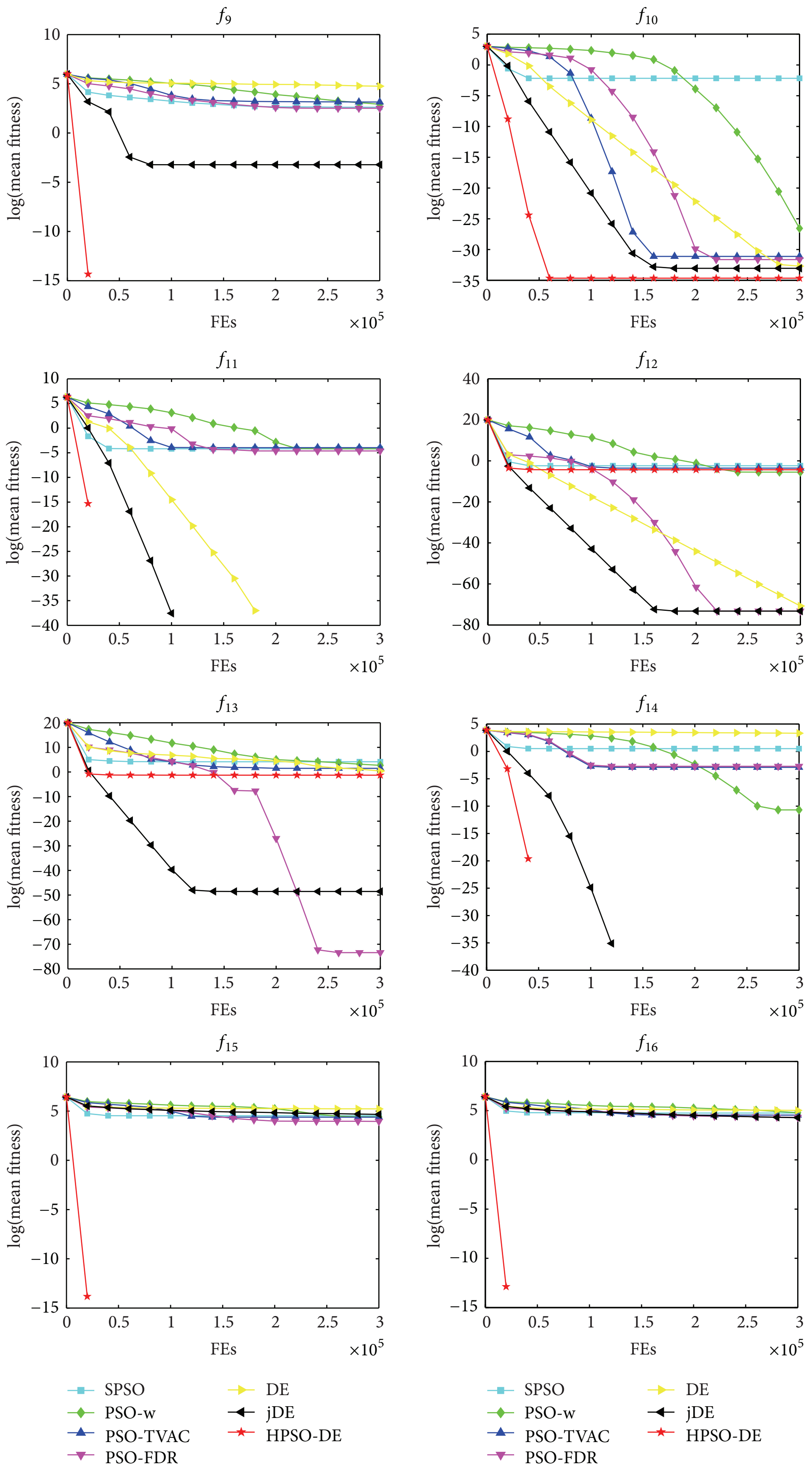


Figure 2: Performance of the algorithms for $f_9$, $f_{10}$ $f_{11}$, $f_{12}$, $f_{13}$, $f_{14}$, $f_{15}$, and $f_{16}$.

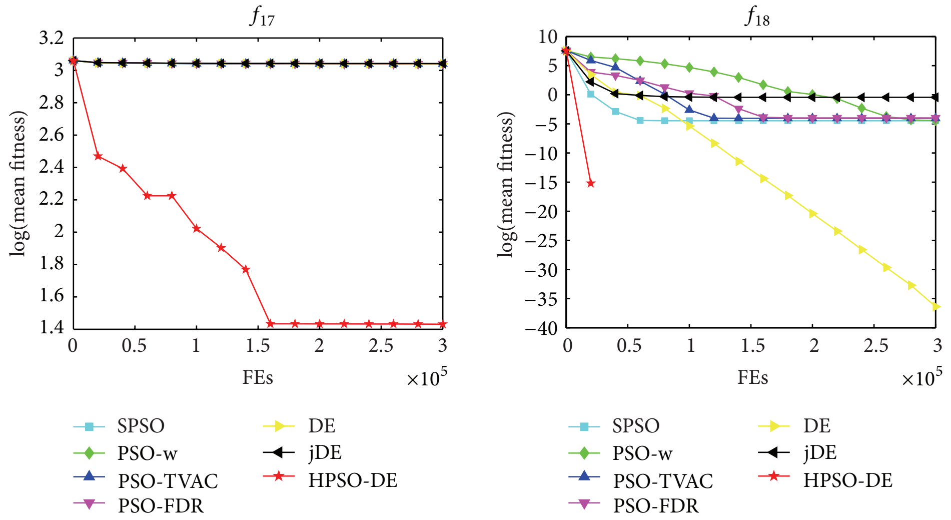


FIGURE 3: Performance of the algorithms for $f_{17}$ and $f_{18}$.

TABLE 1: Benchmark configurations.

| Function | Name | Search space | Global optimal $f(x^*)$ | $x^*$ |
|---|---|---|---|---|
| $f_1$ | Sphere | $[-100, 100]$ | 0 | 0 |
| $f_2$ | Weighted sphere | $[-100, 100]$ | 0 | 0 |
| $f_3$ | Schwefel's Problem 2.22 | $[-10, 10]$ | 0 | 0 |
| $f_4$ | Shifted Schwefel's Problem 1.2 | $[-100, 100]$ | 0 | 0 |
| $f_5$ | Schwefel's Problem 2.21 | $[-100, 100]$ | 0 | 0 |
| $f_6$ | Rosenbrock | $[-30, 30]$ | 0 | $(1, 1, \ldots, 1)$ |
| $f_7$ | Quartic | $[-1.28, 1.28]$ | 0 | 0 |
| $f_8$ | Rastrigin | $[-5, 5]$ | 0 | 0 |
| $f_9$ | Noncontinuous Rastigin | $[-5, 5]$ | 0 | 0 |
| $f_{10}$ | Ackley | $[-32, 32]$ | 0 | 0 |
| $f_{11}$ | Griewank | $[-600, 600]$ | 0 | 0 |
| $f_{12}$ | Penalized 1 | $[-50, 50]$ | 0 | 0 |
| $f_{13}$ | Penalized 2 | $[-50, 50]$ | 0 | 0 |
| $f_{14}$ | Weierstrass | $[-0.5, 0.5]$ | 0 | 0 |
| $f_{15}$ | Rotated Rastrigin | $[-5, 5]$ | 0 | 0 |
| $f_{16}$ | Rotated noncontinuous Rastrigin | $[-5, 5]$ | 0 | 0 |
| $f_{17}$ | Rotated Ackley | $[-32, 32]$ | 0 | 0 |
| $f_{18}$ | Rotated Griewank | $[-600, 600]$ | 0 | 0 |

TABLE 2: Algorithms initialization.

| Algorithm | Parameters | Reference |
|---|---|---|
| SPSO | $w = 0.729$, $c_1 = c_2 = 1.49$ | [41] |
| PSO-w | $w = 0.9 : 0.4$, $c_1 = c_2 = 2$ | [17] |
| PSO-TVAC | $w = 0.9 : 0.4$, $c_1 = 2.5 : 0.5$, $c_2 = 0.5 : 2.5$ | [16, 38] |
| PSO-FDR | $c_1 = c_2 = 1$, $c_3 = 2$ | [42] |
| DE | $F = 0.5$, $\mathrm{CR} = 0.9$ | [20, 43] |
| jDE | $F_l = 0.1$, $F_u = 0.9$, $\tau_1 = \tau_2 = 0.1$ | [29] |
| HPSO-DE | $\mathrm{PSO}_p = 0.3$, $\mathrm{DE}_p = 0.01$, $d_c = 1.5$, $p = 0$ | This paper |

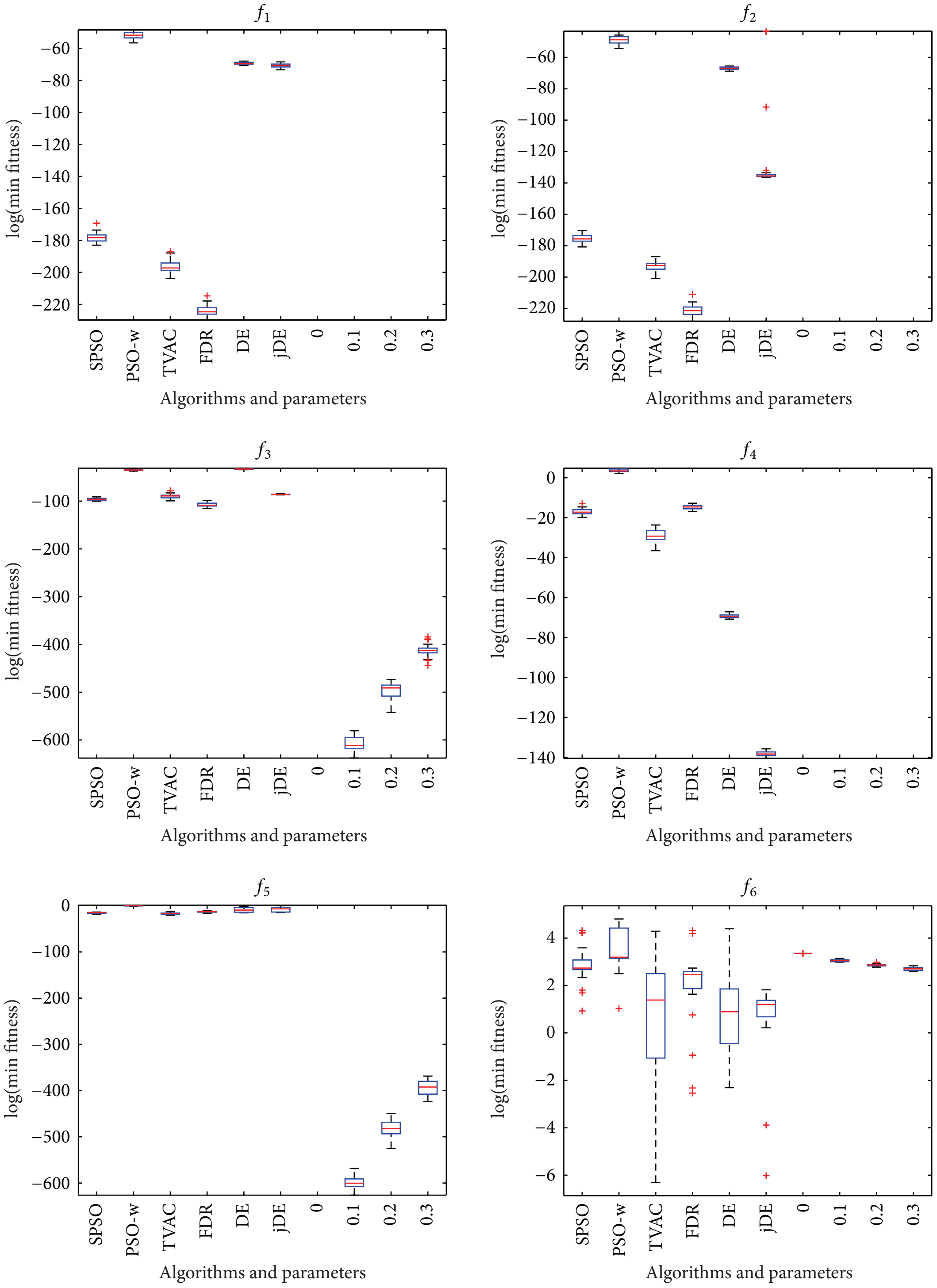


Figure 4: HPSO-DE with different balance parameters and other six algorithms on $f_1$, $f_2$, $f_3$, $f_4$, $f_5$, and $f_6$.

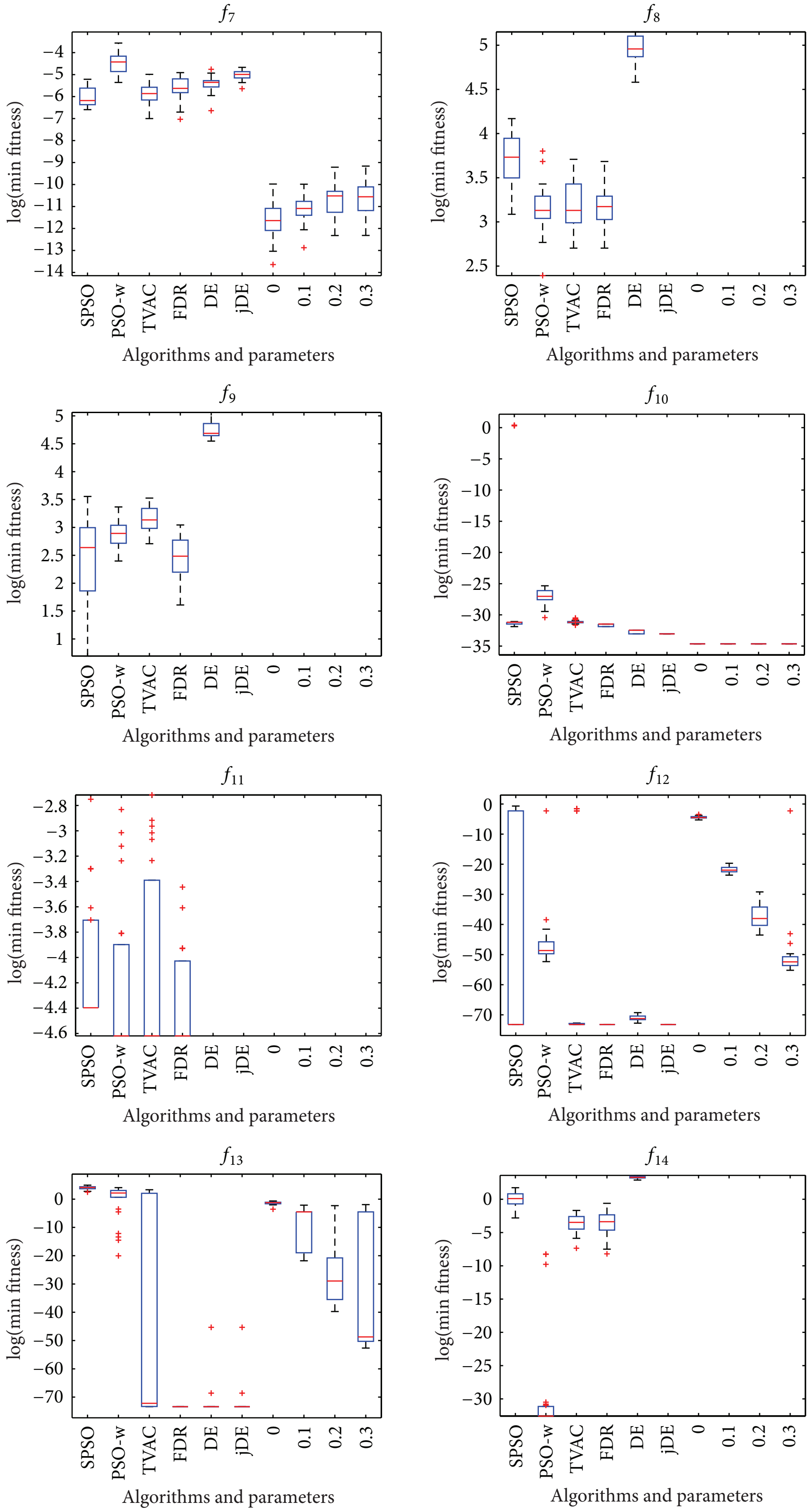


Figure 5: HPSO-DE with different balance parameters and other six algorithms on $f_7$, $f_8$, $f_9$, $f_{10}$, $f_{11}$, $f_{12}$, $f_{13}$, and $f_{14}$.

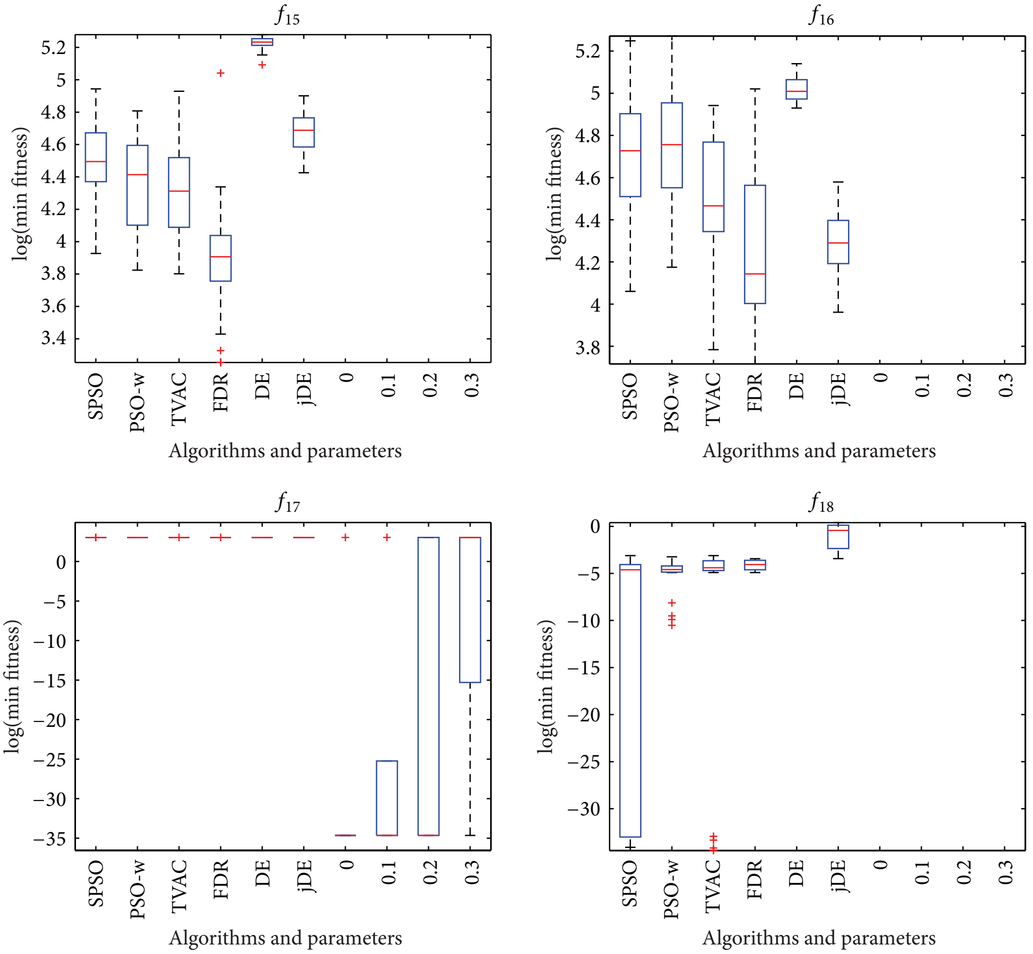


FIGURE 6: HPSO-DE with different balance parameters and other six algorithms on $f_{15}$, $f_{16}$, $f_{17}$, and $f_{18}$.

$$f_{10} = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right) + 20 + e,$$

$$f_{11} = \frac{1}{4000}\sum_{i=1}^{D}x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{i^{1/2}}\right) + 1,$$

$$f_{12}\left(x\right) = \frac{\pi}{n}\left\{10\sin^2\left(\pi y_i\right) + \sum_{i=1}^{D}\left(y_i - 1\right)^2\left[1 + 10\sin^2\left(\pi y_{i+1}\right)\right] + \left(y_n - 1\right)^2\right\} + \sum_{i=1}^{D}u\left(x_i, 10, 100, 4\right),$$

$$y_i = \left(1 + \frac{1}{4}\left(x_i + 1\right)\right),$$

$$u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k\left(-x_i - a\right)^m, & x_i < -a, \end{cases}$$

$$f_{13}\left(x\right) = 0.1\left\{\sin^2\left(3\pi x_1\right) + \sum_{i=1}^{D-1}\left(x_i - 1\right)^2\left[1 + \sin^2\left(3\pi x_{i+1}\right)\right] + \left(x_D - 1\right)^2\left[1 + \sin^2\left(2\pi x_D\right)\right]\right\} + \sum_{i=1}^{D}u\left(x_i, 10, 100, 4\right),$$

Table 3: Result comparisons among seven algorithms on test functions.

| Functions | | Algorithms | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSO | PSO-w | PSO-FDR | PSO-TVAC | DE | jDE | HPSO-DE |
| $f_1$ | Mean | 1.31E − 75 | 1.61E − 22 | 2.39E − 95 | 4.14E − 83 | 1.04E − 30 | 3.84E − 31 | **0.0E + 00** |
| | Std. | 4.03E − 149 | 7.47E − 44 | 1.29E − 188 | 1.62E − 164 | 6.81E − 61 | 2.14E − 61 | **0.0E + 00** |
| $f_2$ | Mean | 7.44E − 76 | 2.80E − 21 | 9.28E − 94 | 4.76E − 83 | 1.33E − 29 | 5.88E − 21 | **0.0E + 00** |
| | Std. | 4.35E − 150 | 1.47E − 41 | 2.09E − 185 | 2.0E − 164 | 1.36E − 58 | 8.63E − 40 | **0.0E + 00** |
| $f_3$ | Mean | 2.50E − 41 | 1.99E − 15 | 6.10E − 45 | 3.88E − 36 | 7.51E − 15 | 5.57E − 038 | **0.0E + 00** |
| | Std. | 3.36E − 81 | 1.15E − 29 | 6.64E − 88 | 3.68E − 70 | 5.05E − 29 | 1.45E − 75 | **0.0E + 00** |
| $f_4$ | Mean | 1.83E − 7 | 4.70E + 1 | 6.77E − 7 | 5.74E − 12 | 1.17E − 30 | 1.80E − 60 | **0.0E + 00** |
| | Std. | 2.21E − 13 | 9.39E + 2 | 5.42E − 13 | 1.80E − 22 | 1.92E − 60 | 6.01E − 120 | **0.0E + 00** |
| $f_5$ | Mean | 2.63E − 7 | 1.0E + 0 | 4.26E − 6 | 1.91E − 7 | 1.59E − 2 | 5.92E − 2 | **0.0E + 00** |
| | Std. | 5.79E − 14 | 3.0E − 1 | 5.12E − 11 | 2.52E − 13 | 8.59E − 4 | 3.31E − 2 | **0.0E + 00** |
| $f_6$ | Mean | 2.24E + 1 | 4.89E + 1 | 1.41E + 1 | 1.10E + 1 | 1.10E + 1 | **3.04E + 0** | 2.86E + 2 |
| | Std. | 3.69E + 2 | 1.53E + 3 | 3.14E + 2 | 3.50E + 2 | 5.09E + 2 | 2.39E + 0 | **9.0E − 3** |
| $f_7$ | Mean | 2.0E − 3 | 1.2E − 2 | 4.0E − 3 | 3.0E − 3 | 4.0E − 3 | 7.0E − 3 | **1.2E − 5** |
| | Std. | 1.16E − 6 | 2.60E − 5 | 3.18E − 6 | 2.01E − 6 | 2.33E − 6 | 1.70E + 0 | **1.049E − 10** |
| $f_8$ | Mean | 4.31E + 1 | 2.44E + 1 | 2.45E + 1 | 2.57E + 1 | 1.42E + 2 | **0.0E + 00** | **0.0E + 00** |
| | Std. | 1.31E + 2 | 4.94E + 1 | 3.27E + 1 | 6.10E + 1 | 5.11E + 2 | **0.0E + 00** | **0.0E + 00** |
| $f_9$ | Mean | 1.37E + 1 | 1.86E + 1 | 1.24E + 1 | 2.33E + 1 | 1.16E + 2 | 4.0E − 2 | **0.0E + 00** |
| | Std. | 7.75E + 1 | 1.90E + 1 | 1.61E + 1 | 2.43E + 1 | 2.82E + 2 | 4.0E − 2 | **0.0E + 00** |
| $f_{10}$ | Mean | 1.14E − 1 | 2.92 − E12 | 1.82E − 14 | 3.08E − 14 | 6.29E − 15 | 4.44E − 15 | **8.882E − 16** |
| | Std. | 1.55E − 1 | 6.88E − 24 | 1.30E − 29 | 8.60E − 29 | 3.28E − 30 | **0.0E + 00** | **0.0E + 00** |
| $f_{11}$ | Mean | 1.5E − 2 | 1.5E − 2 | 1.0E − 2 | 1.9E − 2 | **0.0E + 00** | **0.0E + 00** | **0.0E + 00** |
| | Std. | 2.44E − 4 | 3.0E − 4 | 8.97E − 5 | 4.40E − 4 | **0.0E + 00** | **0.0E + 00** | **0.0E + 00** |
| $f_{12}$ | Mean | 9.1E − 2 | 4.0E − 3 | **1.57E − 32** | 2.90E − 2 | 1.87E − 31 | 1.58E − 32 | 1.3E − 2 |
| | Std. | 3.2E − 2 | 4.0E − 4 | **3.12E − 95** | 4.0E − 3 | 3.39E − 62 | **3.12E − 95** | 3.30E − 5 |
| $f_{13}$ | Mean | 6.34E + 1 | 1.45E + 1 | **1.35E − 32** | 4.44E + 0 | 1.49E + 0 | 8.33E − 22 | 2.76E − 1 |
| | Std. | 1.07E + 3 | 2.13E + 2 | **3.12E − 95** | 5.26E + 1 | 1.68E + 1 | 1.73E − 41 | 1.2E − 2 |
| $f_{14}$ | Mean | 1.67E + 0 | 2.28E − 5 | 6.65E − 2 | 5.5E − 2 | 2.79E + 1 | **0.0E + 00** | **0.0E + 00** |
| | Std. | 2.03E + 0 | 5.10E − 9 | 1.18E − 2 | 3.0E − 3 | 2.78E + 1 | **0.0E + 00** | **0.0E + 00** |
| $f_{15}$ | Mean | 9.25E + 1 | 8.24E + 1 | 5.23E + 1 | 8.01E + 1 | 1.86E + 2 | 1.07E + 2 | **0.0E + 00** |
| | Std. | 4.84E + 2 | 5.01E + 2 | 5.99E + 2 | 6.77E + 2 | 5.82E + 1 | 1.97E + 2 | **0.0E + 00** |
| $f_{16}$ | Mean | 1.17E + 2 | 1.20E + 2 | 7.52E + 1 | 9.38E + 1 | 1.52E + 2 | 7.29E + 1 | **0.0E + 00** |
| | Std. | 1.33E + 3 | 1.00E + 3 | 8.81E + 2 | 7.02E + 2 | 9.76E + 1 | 1.28E + 2 | **0.0E + 00** |
| $f_{17}$ | Mean | 2.09E + 1 | 2.09E + 1 | 2.09E + 1 | 2.09E + 1 | 2.09E + 1 | 2.10E + 1 | **4.18E + 0** |
| | Std. | 2.6E − 3 | 4.3E − 3 | 2.2E − 3 | 2.1E − 3 | **1.80E − 3** | 2.2E − 3 | 7.29E + 1 |
| $f_{18}$ | Mean | 1.15E − 2 | 1.17E − 2 | 1.87E − 2 | 1.76E − 2 | 1.60E − 16 | 6.47E − 1 | **0.0E + 00** |
| | Std. | 1E − 4 | 1E − 4 | 1E − 4 | 2E − 4 | 2.06E − 31 | 2.74E − 1 | **0.0E + 00** |

$$f_{14} = \sum_{i=1}^{D} \left( \sum_{k=0}^{k_{\max}} \left[ a^k \cos\left(2\pi b^k \left(x_i + 0.5\right)\right)\right] - D \sum_{k=0}^{k_{\max}} \left[ a^k \cos\left(2\pi b^k * 0.5\right)\right] \right),$$

$$a = 0.5, \quad b = 3, \quad k_{\max} = 20,$$

$$f_{15} = \sum_{i=1}^{D} \left[ y_i^2 - 10\cos\left(2\pi y_i\right) + 10 \right], \quad y = M \times x,$$

$$f_{16}(x) = \sum_{i=1}^{D} \left( z_i^2 - 10\cos\left(2\pi z_i\right) + 10 \right),$$

$$z_i = \begin{cases} y_i, & \left|y_i\right| < 0.5, \\ \dfrac{\operatorname{round}\left(2y_i\right)}{2}, & \left|y_i\right| \geq 0.5, \end{cases} \quad y = M \times x,$$

$$f_{17} = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi y_i\right)\right) + 20 + e,$$

$$y = M \times x,$$

$$f_{18} = \frac{1}{4000}\sum_{i=1}^{D} y_i^2 - \prod_{i=1}^{D}\cos\left(\frac{y_i}{i^{1/2}}\right) + 1, \quad y = M \times x. \tag{19}$$

Table 4: Convergence speed and algorithm reliability comparisons.

| Functions | | | | | Algorithms | | | |
|---|---|---|---|---|---|---|---|---|
| | | PSO | PSO-w | PSO-FDR | PSO-TVAC | DE | jDE | HPSO-DE |
| $f_1$ | FESS | 4.5056E + 4 | 2.3642 + E5 | 1.4738E + 5 | 1.0142E + 5 | 1.1055E + 5 | 1.0883E + 5 | **2.2075E + 4** |
| | SR | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** |
| $f_2$ | FESS | 4.9428E + 4 | 2.4217E + 5 | 1.5194E + 5 | 1.0435E + 5 | 1.1905E + 5 | 6.4498E + 4 | **2.3002E + 4** |
| | SR | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** |
| $f_3$ | FESS | 6.8336E + 4 | 2.5089E + 5 | 1.6382E + 5 | 1.1524E + 5 | 1.8979E + 5 | 7.6644E + 4 | **3.2762E + 4** |
| | SR | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** |
| $f_4$ | FESS | 2.907E + 5 | — | — | 2.4757E + 5 | 1.1041E + 5 | 5.9383E + 4 | **2.2075E + 4** |
| | SR | 12% | 0% | 0% | **100%** | **100%** | **100%** | **100%** |
| $f_5$ | FESS | 2.987E + 5 | — | — | 2.8725E + 5 | — | — | **3.2629E + 4** |
| | SR | 4% | 0% | 0% | 24% | 0% | 0% | **100%** |
| $f_8$ | FESS | — | — | — | — | — | 7.3477E + 4 | **2.3084E + 4** |
| | SR | 0% | 0% | 0% | 0% | 0% | 100% | **100%** |
| $f_9$ | FESS | — | | — | — | — | 8.7133E + 4 | **2.2356E + 4** |
| | SR | 0% | | 0% | 0% | 0% | 96% | **100%** |
| $f_{10}$ | FESS | 7.337E + 4 | 2.7072E + 5 | 1.7226E + 5 | 1.2118E + 5 | 1.7113E + 5 | 9.0429E + 4 | **3.2502E + 4** |
| | SR | 92% | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** |
| $f_{11}$ | FESS | 4.6325E + 4 | 2.3907E + 5 | 1.5296E + 5 | 1.0411E + 5 | 1.1398E + 5 | 6.2468E + 4 | 2.0659E + 4 |
| | SR | 32% | 28% | 36% | 36% | **100%** | **100%** | **100%** |
| $f_{12}$ | FESS | 4.6978E + 4 | 2.4440E + 5 | 1.3834E + 5 | 1.0872E + 5 | 1.0199E + 5 | **5.0201E + 4** | — |
| | SR | 72% | 96% | **100%** | 80% | **100%** | **100%** | 0% |
| $f_{13}$ | FESS | — | 2.975E + 5 | 1.6684E + 5 | 1.4981E + 5 | 2.5882E + 5 | **5.7271E + 4** | — |
| | SR | 0% | 4% | **100%** | 60% | 52% | **100%** | 0% |
| $f_{14}$ | FESS | — | 2.7346E + 5 | — | — | — | 8.435E + 4 | **3.85E+4** |
| | SR | 0% | 88% | 0% | 0% | 0% | **100%** | **100%** |
| $f_{15}$ | FESS | — | — | — | — | — | — | **2.1025E + 4** |
| | SR | 0% | 0% | 0% | 0% | 0% | 0% | **100%** |
| $f_{16}$ | FESS | — | — | — | — | — | — | **2.2563E + 4** |
| | SR | 0% | 0% | 0% | 0% | 0% | 0% | **100%** |
| $f_{17}$ | FESS | — | — | — | — | — | — | **6.8878E + 4** |
| | SR | 0% | 0% | 0% | 0% | 0% | 0% | **80%** |
| $f_{18}$ | FESS | 1.0931E + 5 | — | — | 1.4448E + 5 | 1.834E + 5 | — | **2.268E + 4** |
| | SR | 36% | 0% | 0% | 20% | 100% | 0% | **100%** |

*4.2. Algorithms for Comparison.* Experiments are conducted on a suite of 16 numerical functions to evaluate seven algorithms including the proposed HPSO-DE algorithm. For functions, 30-dimensional (30D) function is tested. The maximum number of function evaluations (FEs) is set to 300 000 and NP is 100. All experiments are run 25 times independently. The seven algorithms in comparison are listed in Table 2.

*4.3. Comparisons on the Solution Accuracy.* The mean and standard deviation (Std) of the solutions in 25 independent runs are listed in Table 3. The best result among these algorithms is indicated by boldface in the table. Figures 1, 2, and 3 show the comparisons in terms of convergence, mean solutions, and evolution processes in solving 16 benchmark functions.

From Table 3 and Figures 1–3, it is very clear that the hybrid proposed algorithm has the strong ability to jump out of the local optima. It can effectively prevent the premature convergence and significantly enhance the convergence rate and accuracy. It provides best performance on the $f_1$, $f_2$, $f_3$, $f_4$, $f_5$, $f_8$, $f_9$, $f_{11}$, $f_{14}$, $f_{15}$, $f_{16}$, and $f_{18}$, which reachse the highest accuracy on them. The jDE ranks the second on $f_8$, and $f_{11}$ and performs a little better than HPSO-DE on $f_6$. PSO-FDR performs best on $f_{12}$, $f_{13}$.

One can observe that the proposed method can search the optimum and maintain a higher convergence speed. The capabilities of avoiding local optima and finding global optimum of these functions indicate the superiority of HPSO-DE.

*4.4. Comparisons on Convergent Rate and Successful Percentage.* The convergent rate for achieving the global optimum is another key point for testing the algorithm performance. The success of an algorithm means that this algorithm can result

Table 5: Results of the comparison between HPSO-DE and JADE.

| Functions | Gen | HPSO-DE<br>Mean (Std Dev.) | JADE w/o archive<br>Mean (Std Dev.) | JADE with archive<br>Mean (Std Dev.) |
|---|---|---|---|---|
| $f_1$ | 1500 | **2.9E − 121 (1.0E − 240)** | 1.8E − 60 (8.4E − 60) | 1.3E − 54 (9.2E − 54) |
| $f_3$ | 2000 | **1.7E − 78 (4.5E − 155)** | 1.8E − 25 (8.8E − 25) | 3.9E − 22 (2.7E − 21) |
| $f_4$ | 5000 | **0.0E + 00 (0.0E + 00)** | 5.7E − 61 (2.7E − 60) | 6.0E − 87 (1.9E − 86) |
| $f_5$ | 5000 | **1.7E − 185 (00.0E + 00)** | 8.2E − 24 (4.0E − 23) | 4.3E − 66 (1.2E − 65) |
| $f_6$ | 3000 | 2.9 + E2 (1.1E − 2) | **8.0E − 02 (5.6E − 01)** | 3.2E − 01 (1.1E + 00) |
|  | 20000 | 2.9 + E2 (3.4E − 2) | **8.0E − 02 (5.6E − 01)** | 3.2E − 01 (1.1E + 00) |
| $f_7$ | 3000 | **2.1E − 5 (2.2E − 10)** | 6.4E − 04 (2.5E − 04) | 6.8E − 04 (2.5E − 04) |
| $f_8$ | 1000 | **0.0E + 00 (0.0E + 00)** | 1.0E − 04 (6.0E − 05) | 1.4E − 04 (6.5E − 05) |
|  | 5000 | **0.0E + 00 (0.0E + 00)** | **0.0E + 00 (0.0E + 00)** | **0.0E + 00 (0.0E + 00)** |
| $f_{10}$ | 500 | **8.9E − 16 (0.0E + 00)** | 8.2E − 10 (6.9E − 10) | 3.0E − 09 (2.2E − 09) |
|  | 2000 | **8.9E − 16 (0.0E + 00)** | 4.4E − 15 (0.0E + 00) | 4.4E − 15 (0.0E + 00) |
| $f_{11}$ | 500 | **0.0E + 00 (0.0E + 00)** | 9.9E − 08 (6.0E − 07) | 2.0E − 04 (1.4E − 03) |
|  | 3000 | **0.0E + 00 (0.0E + 00)** | **0.0E + 00 (0.0E + 00)** | 2.0E − 04 (1.4E − 03) |
| $f_{12}$ | 500 | 1.8E − 2 (1E − 4) | **4.6E − 17 (1.9E − 16)** | 3.8E − 16 (8.3E − 16) |
|  | 1500 | 1.7E − 2 (4E − 4) | **1.6E − 32 (5.5E − 48)** | **1.6E − 32 (5.5E − 48)** |
| $f_{13}$ | 500 | 4.4E − 1 (2.1E − 2) | **2.0E − 16 (6.5E − 16)** | 1.2E − 15 (2.8E − 15) |
|  | 1500 | 3.1E − 1 (1.6E − 2) | **1.4E − 32 (1.1E − 47)** | **1.4E − 32 (1.1E − 47)** |

in a function value not worse than the prespecified optimal value, that is, for all problems with the number of function evaluations less than the pre specified maximum number. The success rate (SR) is calculated as the number of successful runs divided by the total number of runs.

In Table 4, we summarize the SR of each algorithm and the average number of function evaluations over successful runs (FESS). An experiment is considered as successful if the best solution is found with sufficient accuracy: $10^{-8}$.

Table 4 shows that HPSO-DE needs least FESS to achieve the acceptable solution on most of functions, which reveals that proposed algorithm has a higher convergent rate than other algorithms. DE and jDE outperform HPSO-DE on the $f_{12}$ and $f_{13}$; SPSO, LPSO, and PSO-TVAC have much worse SR and accuracy than HPSO-DE on the test functions. In addition, HPSO-DE can achieve accepted value with a good convergence speed and accuracy on most of the functions, as seen from Figures 1–3 and Table 3.

In summary, the HPSO-DE performs best on functions and has good search ability. Owing to the proposed techniques, the HPSO-DE processes capabilities of fast convergence speed, the highest successful rate, and the best search accuracy among these algorithms.

*4.5. Parameter Study.* The balanced parameter $p$ needs to be optimized. In this section, we investigate the impact of this parameter on HPSO-DE. The HPSO-DE algorithm runs 25 times on each function with four different balanced parameters of 0, 0.1, 0.2, and 0.3. The influence of balanced parameters on accuracy of HPSO-DE algorithm is investigated by comparing the optima values that HPSO-DE obtains for different balanced parameters.

Figures 4, 5, and 6 show the box plots of minimal values that HPSO-DE obtains with four different balanced parameters. The box has lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the remaining data. Outliers are data with values beyond the ends of the whiskers.

From Figures 4–6, one can observe that the accuracy of HPSO-DE is less sensitive to the balanced parameter on most of functions except $f_{12}$, $f_{13}$, and $f_{17}$ when balanced parameter is between 0 and 0.3.

*4.6. Comparison with JADE.* The JADE algorithm is tested on a set of standard test functions in [31]. HPSO-DE is compared with JADE on 30D test functions chosen from [31]. The parameter settings are the same as in [31]. Maximum generations are listed in Table 5. The middle results of 50 independent runs are summarized in the table (results for JADE are taken from [31]), which show that the proposed algorithm obviously performs better than the JADE algorithm.

## 5. Conclusions

In this paper, a novel algorithm HPSO-DE is proposed by developing a balanced parameter between PSO and DE. The population is generated either by DE or by PSO according to the balanced parameter. Adaptive mutation is carried out to current population when the population clusters around local optima. The strategy makes HPSO-DE have the advantages of two algorithms and maintain diversity of the population. In comparison with the PSO, DE, and their variants, the proposed algorithm is more effective in obtaining better quality solutions, works in a more effective way, and finds better quality solutions more frequently.

## 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 Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), Social Science Foundation of Chinese Ministry of Education (nos. 12YJC630271 and 13YJC630018), Natural Science Fund for Colleges and Universities in Jiangsu Province (no. 13KJB120008), China Natural Science Foundation (nos. 71273139, 71101073 and 71173116) and China Institute of Manufacturing Development (no. SK20130090-15).

## References

[1] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[2] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, "Bare bones differential evolution," *European Journal of Operational Research*, vol. 196, no. 1, pp. 128–139, 2009.

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.

[4] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, 1995.

[5] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[6] Y. Tang, Z. Wang, H. Gao, S. Swift, and J. Kurths, "A constrained evolutionary computation method for detecting controlling regions of cortical networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 6, pp. 1569–1581, 2012.

[7] Y. Tang, Z. Wang, and J. Fang, "Controller design for synchronization of an array of delayed neural networks using a controllable probabilistic PSO," *Information Sciences*, vol. 181, no. 20, pp. 4715–4732, 2011.

[8] Y. Tang, Z. Wang, and J. Fang, "Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2523–2535, 2011.

[9] S. Y. S. Leung, Y. Tang, and W. K. Wong, "A hybrid particle swarm optimization and its application in neural networks," *Expert Systems with Applications*, vol. 39, no. 1, pp. 395–405, 2012.

[10] Y. P. Chen, W. C. Peng, and M. C. Jian, "Particle swarm optimization with recombination and dynamic linkage discovery," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 37, no. 6, pp. 1460–1470, 2007.

[11] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[12] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 81–86, Seoul, Republic of Korea, May 2001.

[13] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.

[14] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[15] R. A. Krohling and L. dos Santos Coelho, "Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 36, no. 6, pp. 1407–1416, 2006.

[16] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[17] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pp. 1945–1950, IEEE Press, Piscataway, NJ, USA, 1999.

[18] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the 7th International Conference on Evolutionary Programming VII (LNCS '98)*, pp. 591–600, Springer, New York, NY, USA, 1998.

[19] Z. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 39, no. 6, pp. 1362–1381, 2009.

[20] 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.

[21] J. Liu and J. Lampinen, "On setting the control parameter of the differential evolution method," in *Proceedings of the 8th International Conference on Soft Computing (MENDEL '02)*, pp. 11–18, 2002.

[22] Y. Tang, H. Gao, W. Zou, and J. Kurths, "Identifying controlling nodes in neuronal networks in different scales," *PLoS ONE*, vol. 7, no. 7, Article ID e41375, 2012.

[23] 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.

[24] S. Das and A. Konar, "Automatic image pixel clustering with an improved differential evolution," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 226–236, 2009.

[25] R. Storn, "Differential evolution design of an IIR-filter," in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC '96)*, pp. 268–273, May 1996.

[26] M. Varadarajan and K. S. Swarup, "Differential evolution approach for optimal reactive power dispatch," *Applied Soft Computing Journal*, vol. 8, no. 4, pp. 1549–1561, 2008.

[27] K. Price, R. Storn, and J. Lampinen, *Differential Evolution—A Practical Approach to Global Optimization*, Springer, Berlin, Germany, 2005.

[28] R. Mallipeddi and P. N. Suganthan, “Differential evolution algorithm with ensemble ofparameters and mutation and crossover strategies,” in *Proceedings of the Swarm Evolutionary and Memetic Computing Conference*, vol. 6466, pp. 71–78, Chennai, India, 2010.

[29] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, “Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[30] Z. Yang, K. Tang, and X. Yao, “Self-adaptive differential evolution with neighborhood search,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’08)*, pp. 1110–1116, June 2008.

[31] J. Zhang and A. C. Sanderson, “JADE: adaptive differential evolution with optional external archive,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[32] M. Pant and R. Thangaraj, “DE-PSO: a new hybrid metaheuristic for solving global optimization problems,” *New Mathematics and Natural Computation*, vol. 7, no. 3, pp. 363–381, 2011.

[33] S. Kannan, S. M. R. Slochanal, P. Subbaraj, and N. P. Padhy, “Application of particle swarm optimization technique and its variants to generation expansion planning problem,” *Electric Power Systems Research*, vol. 70, no. 3, pp. 203–210, 2004.

[34] H. Talbi and M. Batouche, “Hybrid particle swarm with differential evolution for multimodal image registration,” in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT ’04)*, vol. 3, pp. 1567–1572, December 2004.

[35] M. G. H. Omran, A. P. Engelbrecht, and A. Salman, “Differential evolution based particle swarm optimization,” in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS ’07)*, pp. 112–119, April 2007.

[36] Z. F. Hao, G. H. Guo, and H. Huang, “A particle swarm optimization algorithm with differential evolution,” in *Proceedings of the 6th International Conference on Machine Learning and Cybernetics (ICMLC ’07)*, pp. 1031–1035, August 2007.

[37] Y. Tang, Z. D. Wang, and J. A. Fang, “Feedback learning particle swarm optimization,” *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 4713–4725, 2011.

[38] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, “Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients,” *Information Sciences*, vol. 177, no. 22, pp. 5033–5049, 2007.

[39] H. Wang, Z. Wu, and S. Rahnamayan, “Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems,” *Soft Computing*, vol. 15, no. 11, pp. 2127–2140, 2011.

[40] H. Sharma, J. C. Bansal, and K. V. Arya, “Fitness based differential evolution,” *Memetic Computing*, vol. 4, pp. 303–316, 2012.

[41] D. Bratton and J. Kennedy, “Defining a standard for particle swarm optimization,” in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS ’07)*, pp. 120–127, April 2007.

[42] T. Peram, K. Veeramachaneni, and C. K. Mohan, “Fitness-distance-ratio based particle swarm optiniization,” in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS ’03)*, pp. 174–181, 2003.

[43] M. M. Ali and A. Törn, “Population set-based global optimization algorithms: some modifications and numerical studies,” *Computers and Operations Research*, vol. 31, no. 10, pp. 1703–1725, 2004.

[44] M. Ali and M. Pant, “Improving the performance of differential evolution algorithm using Cauchy mutation,” *Soft Computing*, vol. 15, no. 5, pp. 991–1007, 2011.

[45] A. K. Qin and P. N. Suganthan, “Self-adaptive differential evolution algorithm for numerical optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’05)*, vol. 2, pp. 1785–1791, September 2005.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 468324, 11 pages
http://dx.doi.org/10.1155/2014/468324

*Research Article*

# A Modified Decision Tree Algorithm Based on Genetic Algorithm for Mobile User Classification Problem

**Dong-sheng Liu**[1,2] **and Shu-jiang Fan**[1]

[1] *College of Computer Science & Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[2] *Center for Studies of Modern Business, Zhejiang Gongshang University, Hangzhou 310018, China*

Correspondence should be addressed to Dong-sheng Liu; lds1118@163.com

Received 31 October 2013; Accepted 23 December 2013; Published 9 February 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

In order to offer mobile customers better service, we should classify the mobile user firstly. Aimed at the limitations of previous classification methods, this paper puts forward a modified decision tree algorithm for mobile user classification, which introduced genetic algorithm to optimize the results of the decision tree algorithm. We also take the context information as a classification attributes for the mobile user and we classify the context into public context and private context classes. Then we analyze the processes and operators of the algorithm. At last, we make an experiment on the mobile user with the algorithm, we can classify the mobile user into Basic service user, E-service user, Plus service user, and Total service user classes and we can also get some rules about the mobile user. Compared to C4.5 decision tree algorithm and SVM algorithm, the algorithm we proposed in this paper has higher accuracy and more simplicity.

## 1. Introduction

With the rapid development of mobile internet, mobile users can enjoy mobile services at anytime from anywhere, such as location-based services, mobile games, location-based advertising, and mobile phone rescue. By the end of March 2013, the number of mobile communication service users in China has reached 1.146 billion, which is 1.24% higher than last month and 12.46% higher than the same period last year. Facing the huge number of users, how to provide personalized services to customers, and how to make customer classification to mobile users based on data mining technologies have become the focus of the current academic and industry attention.

There are many methods which have been used to classify the customer. Han et al. [1] segmented the telecom customers based on customer value by decision tree model, they proposed a novel customer segmentation method based on customer lifecycle, and a decision tree method was developed to extract important parameters of customer value. In this study, the authors only took the customer value into consideration and did not take the social attribute of the user into consideration. Xiao et al. [2] proposed a dynamic classifier ensemble method for imbalanced data. Bayesian network was also used as a tool to the customer classification [3].

In generally, the Bayes classifier is not as sensitive as the C4.5 (one of decision tree algorithm) classifier [4]; compared to neural network, the decision tree has a better quality to deal with the nonnumeric data and can be understand easier; neural network needs many parameters when running it and has a long time to learning [5]; support vector machine classifier has a high precision but the result cannot be understood easily. So we select decision tree as a tool to generate rules in this paper. But most of the decision tree algorithms are greedy algorithm; greedy algorithm is usually running fast, but it does not get the optimal decision tree. To get optimal decision tree problem is NP complete problem; these methods cannot solve it. This paper puts forward a new decision model for mobile user classification, which introduced genetic algorithm to optimize the results of the decision tree algorithm.

## 2. Related Work

*2.1. Classification Model.* There are many classification models which have been proposed by researchers, such as decision tree algorithm, Bayesian network, genetic algorithm, and neural net algorithm.

Zhang [6] took the annual salary, education, age, occupation, marriage, and property attributes of customer as the decision attribute set and established the classification model for Chinese customers of bank based on decision tree. She has classified the customer into risk customer, bad customer, ordinary customer, and important customer classes. She used a single data mining method and used it in bank customer classification. The accuracy of the result may not be very accurate, so it will not be suitable to the mobile user classification.

Chen [7] proposed a tree classification model based on Bayesian network algorithm. This model which the researcher proposed uses a single method to classify the trees, which may be very useful in small data sets. For big data sets, the accuracy of the model will decrease. Moreover, as we mentioned before, the Bayesian classifier is not as sensitive as decision tree classifier.

Zhou et al. proposed a data selection model based on neural network [8]; this model used a modified neural network to constructed the classifier; it may be very useful in reducing time consume, but the accuracy may be not very satisfactory, and the model also used a single method to problem. Moreover as we mentioned before, neural network has a bad quality to deal with the nonnumeric data and low learning rate.

Shu [9] proposed a fingerprint classification system based on a modified genetic algorithm. In this study, an improvement of the born classification is designed by adding a joined BP operator GA; it may suit the fingerprint classification, but it is not very useful to mobile user data.

Zhou has put the applied the SVM in mobile communication churn and got a better result. But as we mentioned before, support vector machine classifier has a high precision but the result cannot be understood easily.

Most of these studies are based on a single data mining technique. There have been few attempts to apply several techniques simultaneously and combine their outcomes for classification model and it is not very useful to mobile user classification.

*2.2. Decision Tree Algorithm.* The classical decision tree algorithm includes ID3 algorithm [10], C4.5 algorithm based on ID3 algorithm [11], CHAID algorithm (CHI-squared Automatic Interaction Detector) [12], and CART algorithm (Classification and Regression Tree) [13].

C4.5 algorithm is a modified algorithm based on ID3. Compared to ID3 algorithm, C4.5 algorithm can describe the continual attribute situation, but ID3 algorithm cannot. And C4.5 algorithm has a faster speed in realizing the process than ID3 algorithm. Moreover the decision tree structure of C4.5 is also more reasonable than ID3 algorithm and also finds the good rules information. Compared to CART decision tree algorithm, C4.5 can construct multitree and CART algorithm only construct binary tree.

As we all know C4.5 is a modified algorithm to generate decision tree based on ID3 algorithm. C5.0/See 5.0 is commercialized versions of C4.5; the core of C5.0/See 5.0 is the same with C4.5, but C5.0/See 5.0 has been modified in execution efficiency and memory. Based on C4.5, C5.0 algorithm not only includes all functions of C4.5, but also introduces many new technologies. Particularly, one of the most important technologies is boosting technique [14, 15] which further improve the recognition rate of the sample. Compared with C4.5, C5.0 with higher accuracy, faster running speed, and smaller decision tree model takes up less computer memory. Additionally, the character of C5.0 algorithm is low complexity, easily and high adaptability. Owning to the advantages of C5.0 algorithm, many scholars have applied the algorithm to a series of applications. For example, based on C5.0 algorithm, Pang and Gong researched personal credit evaluation on bank [16]. Taiwan scholar Chiang [4] classifies clients with C5.0 algorithm.

## 3. User Classification and Mobile User Classification

Normal classification model may not be suitable for the mobile user classification because of dynamic of the mobile user data and because the data is so large. We will analyze this in the follow sections.

*3.1. User Classification.* User classification or customer classification can be defined as verifying the identification and differentiation of customers based on customer attributes. The customer attributes usually include social attributes, behavior, and value attributes. Customer classification can analyze customer's consumption behavior and also can analyze the customer's consumption psychology. Companies can provide different products for different behavior patterns, for different consumer psychology of customers with different promotion methods, and so forth. Customer classification is the basis of the customer analysis, mining the data which are after the classification is more targeted and can get more meaningful results.

In generally, classification methods and cluster methods can be used to user classification. Classification methods, such as decision tree algorithm, neural network, and SVM method. Cluster methods can be described as clustering the user data, analyzing each cluster of the user, and summing up the similarity or some attributes in common in each cluster users.

Mobile user classification may differ with the general users, for it has more attributes, such as context attribute, huge number of user, and we will analysis it in the following section. This requires that the mobile user classification method has higher accuracy, andnormal method we mentioned before will not be suitable for the mobile user classification. So in this paper we proposed a modified decision tree algorithm for the mobile user classification.

*3.2. Mobile User Classification.* There are many works on mobile users, such as Yang and Fei who have researched on Broadcasting in vehicular networks [17] and many classification models for user. But these classification models always have their limitations, such as low accuracy and complex and low running speed.

In the mobile user data set, it always includes a lot of user's attributes, such as basic information about the user: age, income, hometown, education, and so on; other information such as consume information like basic cost and Wireless cost will also contained in the data set; mobile user information also including context information about the mobile user, such as the mobile user can request mobile service with "anytime, any where", it is a dynamic data stream. So if we want to classify the mobile user precisely, we should take the context information into account. In this paper, we can use the context information as a classification attribute. We classify the context attribute into private context and public context classes. In this paper, we can define the private context as the environment information about a mobile user's stays in a private place himself, such as a small room, quiet and whether the user enjoys solitude. The public context can be defined as the environment information about a mobile user's stays in a crowd place; for example, a user stays in a bus station; he may request different services in a private room. In other words, under the different contexts, the user will have different requirements, and it will influence the classification result.

Another important thing is that not every attribute has equally weight to classify the mobile user. Although decision tree could select the main important attributes, the result may not be optimized. In addition, mobile users are a large number in any countries or cities. It can use the "Big data" to express it. So to classify the mobile user with high accuracy, it seems as a hard work to do.

As we analyze above, if we want to classify the mobile user with these numbers of attributes and huge number of users with high accuracy, these models we mentioned before seem not very suitable. So we will propose a new model for the mobile user classification which will classify mobile user into classes with high accuracy.

*3.3. A Modified Mobile User Classification Model.* As mentioned in the previous section, we propose a new customer classification mode based on decision tree and genetic algorithm. The overall framework of the proposed model is shown in Figure 1.

As is shown in Figure 1, the process of this model consists of four steps in total. The detail explanation for each step of the proposed model is presented as follows.

*Step 1. Data Partition.* In this process, we should partition the customer data; we partition the data into training data set and test data set. We can partition the data in percentage terms. For example, if there are $n$ sets of data, we can take $70\% \times n$ of them as the training data and the else of data as the test data; this step will reduce the data amount and will also provide the test data set in the following step.

*Step 2. Generated Rules by Decision Tree.* In this step, we will use decision tree algorithm to the training data set to generate inference rules. The decision tree algorithm can be ID3 algorithm, CHAID algorithm (CHI-squared Automatic Interaction Detector), or C4.5 algorithm. In this paper, we take C4.5 algorithm as a tool to generate rules because of the accuracy and low complexity of the algorithm.

*Step 3. Optimize the Rule by Genetic Algorithm.* After generating rules, we should optimize the rule, because the rule may not bethe optimization for the data. In this paper, we use genetic algorithm to optimize the rule; we will analyze this step in the following section in detail.

*Step 4. Test the Optimized Rule.* In this step, we use the test data set to verify the accuracy of the optimized rule.

Through these steps, we can finally get the optimization rule for the data set. Steps 1, 2, and 4 are normal steps, so we will not describe them again; Step 3 is the main point of our paper, so we will describe it in detail in Section 4.

## 4. Modified Decision Tree for Mobile User Classification Based on Genetic Algorithm

*4.1. The Basic Decision Tree Algorithm.* A decision tree is a flow-chart-like tree structure, where each internal node (nonleaf node) denotes a test on attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node. A typical decision tree is shown in Figure 2. The decision tree algorithm usually has three popular attribute selection measures, namely, information gain, gain ratio, and gini index.

Assuming that $S$ is the set of data samples, the attributes of class label have $m$ different value, and the number of different classes $C_i$ $(i = 1, 2, \ldots, m)$ to be $m$. Set $s_i$ is the number of samples in class $C_i$. For a given sample, the expected information needed for classification is given by the following equation:

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i), \quad (1)$$

where $p_i = s_i/s$ is the probability of any sample belonging to $C_i$.

Set attribute $A$ with $v$ different values $\{a_1, a_2, \ldots, a_v\}$. Then $S$ could be divided into $v$ subsets $\{s_1, s_{2,} \ldots, s_v\}$ by attribute $A$. Where the sample of $s_j$ has the same value $a_j$ $(j = 1, 2, \ldots, v)$ in the attribute $A$. Set $s_{ij}$ to be the number of the sample of class $C_i$ in a subset $s_j$. The entropy and information

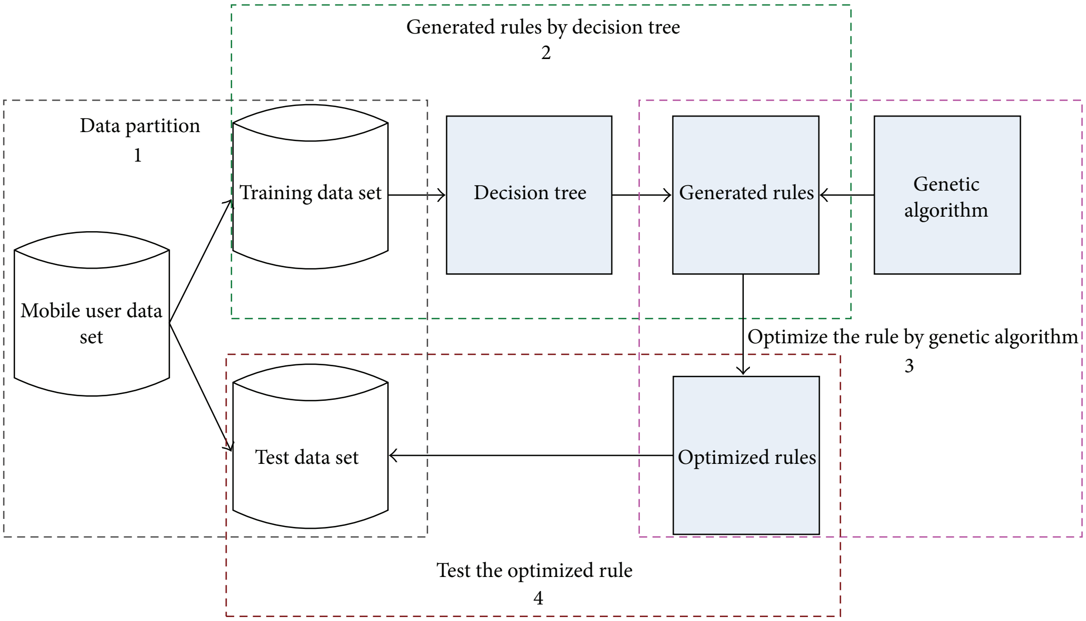


Figure 1: Framework of the proposed model.

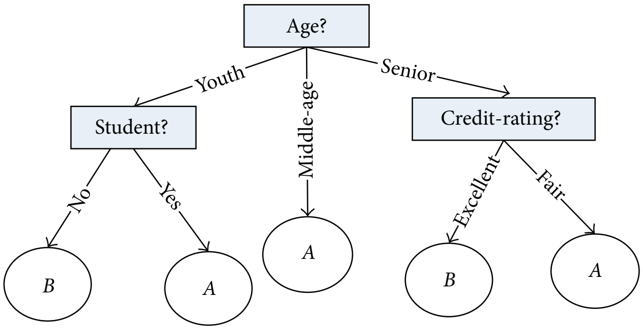


Figure 2: A sample decision tree.

expectations of the subsets divided by $A$ are given by the following expression

$$E(A)=\sum_{i=1}^{v}\frac{\left(s_{1j}+s_{2j}+\cdots+s_{mj}\right)}{s}I\left(s_{1j},s_{2j},\ldots,s_{mj}\right). \quad (2)$$

When the entropy value is smaller, the purity of subset partition will be higher. For a given subset $s_j$, the expected information is:

$$I\left(s_{1j},s_{2j},\ldots,s_{mj}\right)=-\sum_{i=1}^{m}p_{ij}\log_2\left(p_{ij}\right), \quad (3)$$

where $p_{ij} = s_{ij}/s_j$ is the probability of the sample of $s_j$ belonging to $C_i$. If we conduct the branch operation in the attribute $A$, the information gain received is $\text{Gain}(A) = I(s_{1j},s_{2j},\ldots,s_{mj}) - E(A)$. Then according to the split information Split_info($A$) that is used to measure the breadth and uniformity of the split of data, the size of the information gain rate is compared in the process of the attribute classification; then the attribute with the maximum information gain rate is chosen for split attributes.

Where the split information and information gain ratio can be, respectively, expressed as

$$\text{Split\_info}(A)=\sum_{j=1}^{v}\left|\frac{s_j}{s}\right|\log_2\left(\left|\frac{s_j}{s}\right|\right),$$
$$\text{gain\_ration}(A)=\frac{\text{Gain}(A)}{\text{Split\_info}(A)}. \quad (4)$$

Repeat the above steps until all the attributes are classified.

Decision tree cannot only construct the tree but also produce the inference rules. The description is shown as follows.

IF condition 1 and condition 2 and condition 3 . . .

and condition $n$ then Class $A$, (5)

where *condition i* is the preconditions and $A$ is the class type. So we can see that classification rules are logic formulas who come from conjunctive normal form; the left of each rule conjunction item corresponds to the feature attributes. In Figure 2, we can get the following expressions as shown in Algorithm 1.

*4.2. Modified Decision Tree Algorithm Based on Genetic Algorithm.* As we mentioned before, decision tree always cannot get the optimize rule and genetic algorithm is usually used as a optimize tool. So we can use genetic algorithm to optimize the result of decision tree.

The idea of the algorithm we proposed in this paper is that we firstly use the decision tree algorithm to generate the mobile user classification rules, and then according to

IF (*age* = *youth*) and (*student* = *no*) then *Class* = *B*
IF (*age* = *youth*) and (*student* = *yes*) then *Class* = *A*
IF (*age* = *middle-age*) then Class = *A*
IF (*age* = *senior*) and (*credit-rating* = *excellent*) then *Class* = *B*
⋮

Algorithm 1: Expressions for Figure 2.

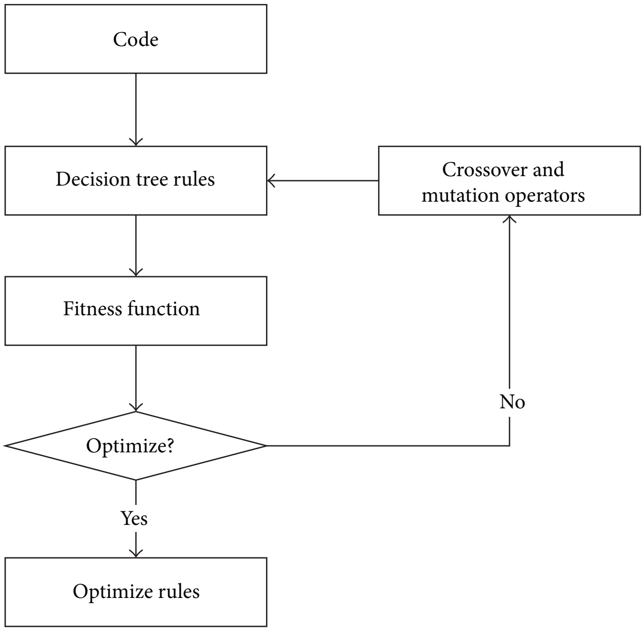


Figure 3: Processes of algorithm.

the attribute of the rule, such as accuracy, support, simplicity, and gain ratio, we construct the fitness function of genetic algorithm. The larger the value of the fitness is, the more the optimal rule will be. We use the crossover operation and mutation operation of genetic to adjust the fitness function, so the fitness value will reach to the maximum value, and the rule will be optimization. The processes are shown in Figure 3. We will describe these steps in following sections.

*4.2.1. Coding for the Rule.* In general, genetic algorithm adopts bit coding with fixed length; the most common of use is the binary code; this method use a string which is constituted by the symbol $\{0, 1\}$ to denote an individual. Each code responding to a condition attribute and the attribute value will determine the encoding length. For example, an attribute has $K$ kinds of value (the continuous attributes need discretized firstly), so the individual coding will distribute $K$ bits for it and each bit corresponding the possible values. When the value is 0, it means that the individual will not take the attribute value. When the value is 1, the individual will take the attribute value. Transformation of this method is simple and each chromosome has fixed length. However, the decision tree has a feature that the node has not only discrete attributes, but also numerical attributes. The simple binary code is not very useful.

In this paper, we set that each chromosome represents a classification rule. Some chromosomes will become the solution of problem. The final rule set will be sorted by the quality of the rule. When the rule set is used to recognize a new sample, the best rule will be considered firstly; if the best rule cannot recognize the sample, then we can choose the next rule. If the rule in the rule set cannot recognize the sample either, the sample will be classified as default class. Chromosomes will compete with each other in priority of the population.

Assume that the data include $n$ attributes, so each chromosome will be divided into $n$ genes, the $i$th gene corresponding to the $i$th attribute. Each individual represents a classification rule and each gene represents the left side of classification rule or the right side. The whole chromosome can represent a completed rule *IF-THEN*. The left side of classification rule is constructed by the genes which correspond to the characteristic attributes; we called these genes as characteristic genes; the right side of the rule is constructed by genes which correspond to class attribute; we called this gene will as class gene. During the gene evolution, the characteristic genes will participate in the evolution, but the class gene not. Each chromosome has a fixed length and has some genes. Inner of each gene includes four parts: {Weight, Operator, Value, Gain ratio}.

*Weight.* Weight is a Boolean variable; it represents weather gene which corresponds to the attribute appears; if the weight is 1, the attribute which corresponds to the gene will appear in the rule. On the contrary, the weight is 0, which means that the attribute which corresponds to the gene will not appear in the rule.

*Operator.* It denotes the operators that genes conjunction adopt. To the discrete attributes, the value should be "=" or "≠"; the continuous attributes, the value, should be "≥" or "<".

*Value.* Value denotes the value of the attribute. To the discrete attributes, the value equals the site where actual values in the domain of value. To the continuous attributes value is equal to the actual value.

*Gain Ratio.* It denotes the information gain rate of the attribute; it can be calculated with the formula in Section 2.

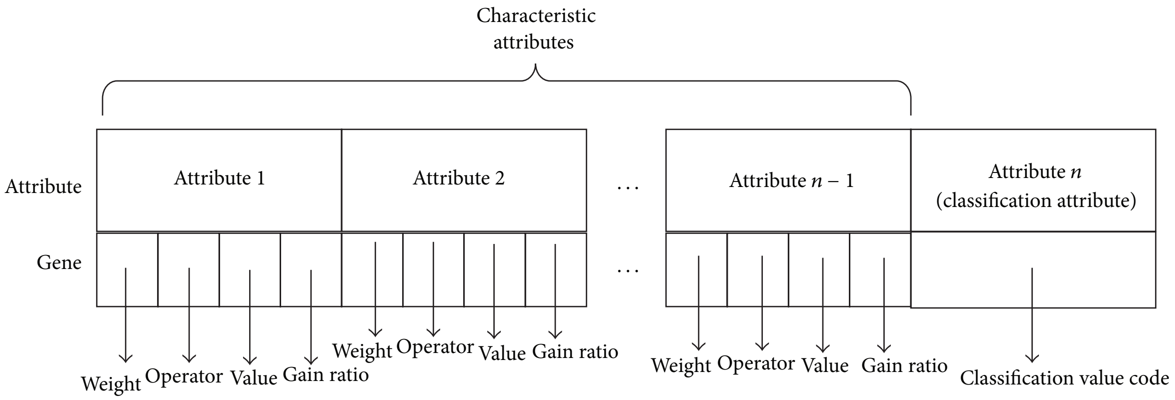


Figure 4: Chromosome construction.

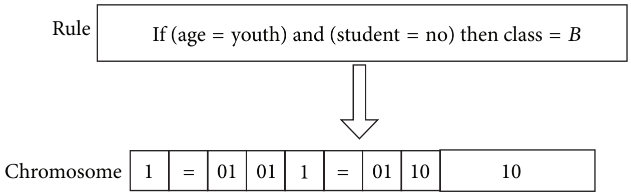


Figure 5: A simple example for rule coding.

Before the genetic algorithm begins, calculate and save all attributes information gain rates in individual. The construction of the chromosome is shown in Figure 4.

In this method, although the length of the chromosome is fixed, the length of rule can be variable; it will mine out rules with more simplicity. As shown in Figure 5, it is a simple example of rule coding; if we have rule IF (*age* = *youth*) and (*student* = *no*) and then *Class* = *B*, then we can code it with "1 = 01011 = 011010."

*4.2.2. Fitness Function for the Rule.* In genetic algorithm, the fitness function is a measure to evaluate good or bad of the individual. In this paper we can divide the sample into four classes:

(1) $T\_T$: it denotes the number of the rule predicting the sample is true and the actual is true;

(2) $T\_F$: it denotes the number of the rule predicting the sample is true and the actual is fault;

(3) $F\_T$: it denotes the number of the rule predicting the sample is fault and the actual is true;

(4) $F\_F$: it denotes the number of the rule predicting the sample is fault and the actual is fault.

As shown before, we can set a variable to construct fitness function, we can call it accuracy. The formula of the accuracy is

$$\text{accuracy} = \frac{T\_T + F\_F}{T + F}, \tag{6}$$

where $T$ is the number of the sample which is true and $F$ is the number of the sample which is fault. The accuracy can be the degree of accuracy the rule works on the training data. The higher the value is, the more samples correct classification.

Another variable is support; the formula of the support is

$$\text{support} = \frac{T\_T + F\_T}{T + F}. \tag{7}$$

The larger the value is, the greater proportion the rule in the data space; it means the rule has a better significance.

In this paper, we set the 3th variable to evaluate the fitness, named simplicity; the formula of the simplicity is

$$\text{simplicity} = \frac{N\,(\text{attributes}) - n\,(\text{rule attributes})}{N\,(\text{attributes})}, \tag{8}$$

where $N(\mathit{attributes})$ is the number of attributes in the data set and $n(\mathit{rules\_attributes})$ is the number of attributes in rule. The higher the simplicity of the individual, the simpler the rule, and the rule can be understood easier.

At last, the genetic algorithm will be used to produce the decision tree, so we should take the information gain ratio as a variable in the fitness function. The information gain ratio can be calculated with the formula as shown in Section 2 and we use gain ratio to express it. As analyzed above, the fitness function can be constructed as in following formula:

$$\begin{aligned}\text{Max fitness} = {} & a \times \text{simplicity} + b \times \text{support} \\ & + c \times \text{accuracy} + d \times \text{Gain Ratio},\end{aligned} \tag{9}$$

where $a$, $b$, $c$, and $d$ are weight of the variables which in $[0, 1]$ and $a + b + c + d = 1$.

*4.2.3. Crossover and Mutation Operations for the Rule.* In this paper, we should select a sample $R$ in training data where classification attribute is $C_i$, randomly, and then code $R$ to the individual coding string based on the code rules. In this way, the new generated individuals are effective individuals; it will reduce the search space of the algorithm greatly and improve the speed of the algorithm.

Two-point crossover is used to the chromosomal chiasma in this paper; firstly produce a random real number $Sc$ which

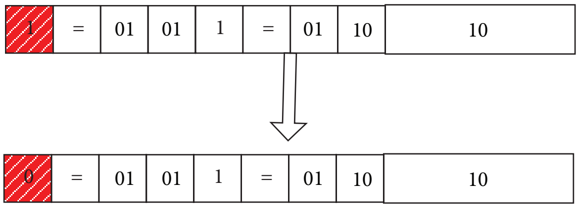


Figure 6: An example for weight mutation.

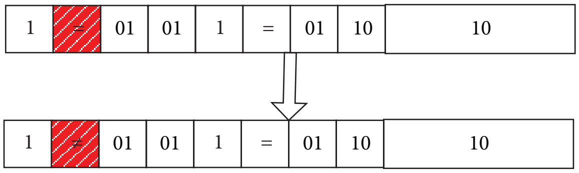


Figure 7: An example for operator mutation.

in [0, 1], if $Sc$ is less than the crossover probability $Pc$, then select individuals $a_i$ and $a_j$ randomly to crossover.

Produce a random real number $Sm$ which in [0, 1], if $Sm$ is less than the mutation probability $P_m$, we will do mutation on the individual. For the gene in this paper having four parts, so we must consider the gene construction sufficiently. So it will include three mutation operations (gain ratio gene will not change in these operations).

*Weight Mutation.* If the weight of original gene is 1, then mutate it to 0; if the weight of original gene is 0, then mutate it to 1. In this paper, we set that if the weight mutates from 1 to 0, the attribute which the gene corresponds to will not appear in the rule. For example, as shown in Figure 6, through the weight mutation, we can get the following rule:

$$\text{IF } (\text{student} = \text{no}) \text{ then Class} = B. \tag{10}$$

*Operator Mutation.* To the discrete attributes, if the operator of original gene is "=," then mutate it to "≠"; if the operator of original gene is "≠," then mutate it to "=." To the continuous attributes, if the operator of original gene is "≥," then mutate it to the "<"; if the operator of original gene is "<," then mutate it to the "≥"; for example, as shown in Figure 7, through the operator mutation, we can get the following rule:

$$\text{IF } (\text{age} \neq \text{youth}) \text{ and } (\text{student} = \text{no}), \text{then Class} = B. \tag{11}$$

That is to say

$$\begin{aligned} &\text{IF } (\text{age} = \text{middle\_age or age} = \text{senior}) \\ &\text{and } (\text{student} = \text{no}), \text{then Class} = B. \end{aligned} \tag{12}$$

*Value Mutation.* To the discrete attributes, choose a value in the attribute to replace the value in the original, randomly; to the continuous attributes, produce a decimal randomly and then do plus or minus on the original value with the decimal. For example, as shown in Figure 8, through value mutation, we can get the following rule:

$$\begin{aligned} &\text{IF } (\text{age} = \text{middle\_age}) \\ &\text{and } (\text{student} = \text{no}), \text{ then Class} = B. \end{aligned} \tag{13}$$

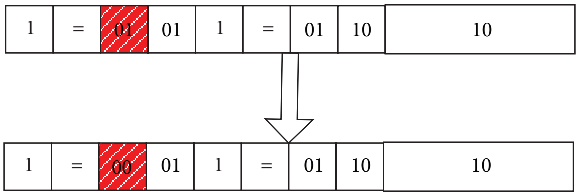


Figure 8: An example for value mutation.

Each mutation can be one mutation operation or any combination of the mutation operations.

*4.3. Algorithm Description.* The whole algorithm process flow can be described as follows.

*Step 1.* Initialize the population; a sample $R$ with $S$ records is randomly selected from the training set, whose class attribute value is $C_i$. Then the evolution algebra variable num and the population initial average fitness variable $avg$ are both assigned zero.

*Step 2.* Preprocessing operations were conducted on the sample $R$, including data cleaning, continuous attribute discretization, calculating the information gain rations of each feature attribute, and encoding the record data. Ultimately we have the initial encoded population $P(r)$.

*Step 3.* Compute the fitness of each individual in the population, and then the average fitness is figured out.

*Step 4.* If the value of num is less than the maximum evolution population or $avg_i - avg_{i-1} > \varepsilon$, then repeat Steps 5, 6, and 7; otherwise go to Step 8.

*Step 5.* Calculating the average fitness of this generation, selection, crossover, and mutation operations are conducted on this population; thus offspring population is generated.

*Step 6.* Replace the individuals with low fitness in the parent by the ones that have high fitness in the offspring population; therefore, new generation is formed.

*Step 7.* Compute the fitness of each individual in the new generation, the average fitness as well.

*Step 8.* Those individuals whose fitness value is less than the lowest fitness threshold are taken out. The optimized population is the optimal set of rules.

The framework of the algorithm processes is shown in Figure 9 and the algorithm description is shown in Algorithm 2.

## 5. Mobile Customer Classification

We will use the modified decision tree algorithm based on genetic algorithm we proposed in this paper to deal with

```
Algorithm: Decision tree_Genetic algorithm
Input:
       Data set R, parameters for Genetic algorithm
Output:
       Optimal classification rules
Begin:
   $I = 0$;
   Initialize $P(I)$;
     // initialize population; Scan R, select records where class attribute value = $C_i$
Preprocessing R;
     //this process including clean data, discretize continuous attributes, calculate each feature attribute information gain rate,
    code recorded data and get initial population.
Fitness $P(I)$;
Avg($Fitness\ P(I)$);
While ($I <= Max$_generation or $\mathrm{Avg}_i - \mathrm{Avg}_{i-1} > \varepsilon$)
{
                 $I$ ++;
                 $GA$-$Operation\ P(I)$;
                 $Fitness\ P(I)$;
          }
Or delete the individual which fitness less than the threshold value;
Optimal Population $P(I)$;
   //get the optimal classification rules and calculate the average the $Fitness(I)$
END
```

Algorithm 2: Algorithm description.

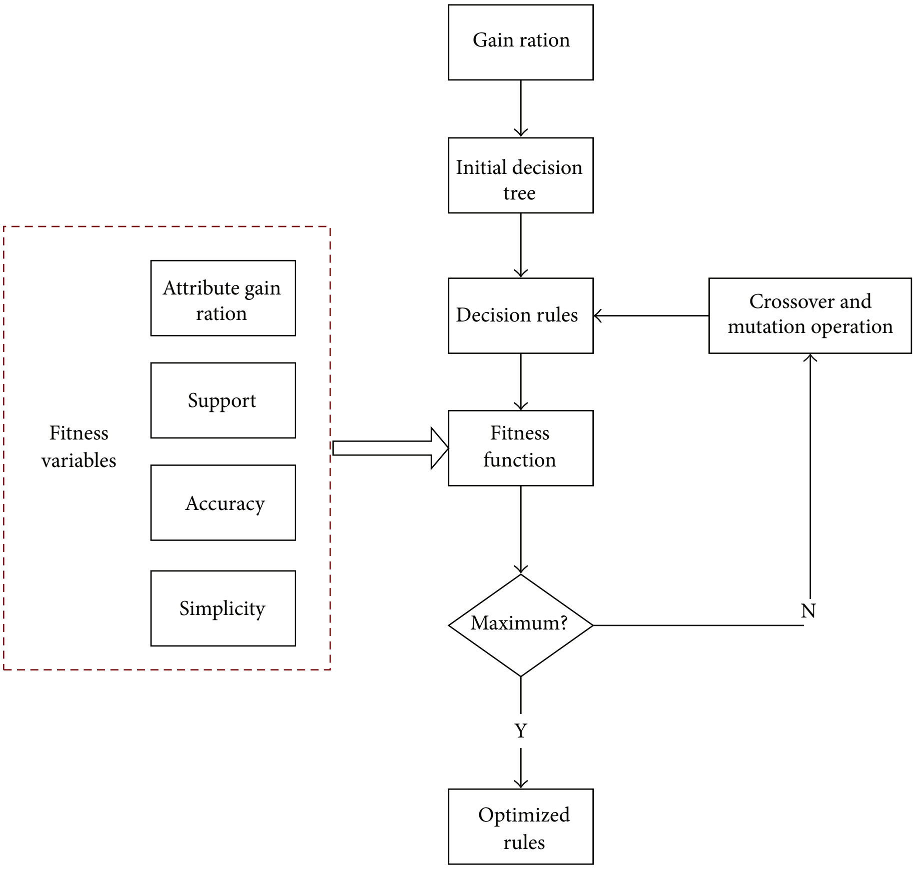


Figure 9: Modified decision tree algorithm processes.

Table 1: Mobile user attributes information.

| Attribute | Comment | Type | Value |
|---|---|---|---|
| Month | How many months did the user use the mobile service in? | Numeric | None |
| Wireless service | Did the user use the wireless service? | Boolean | 0: no<br>1: yes |
| Basic cost | How much did the user spend on the mobile service? | Numeric | None |
| Free part | How much was the service to the user for free? | Numeric | None |
| Wireless cost | How much did the user spend on the wireless service? | Numeric | None |
| Context information | Which context is the user in? | Discrete | Private context<br>Public context |
| Class | The class which the mobile user belong to. | Discrete | Basic service<br>E-service<br>Plus service<br>Total service |

Table 2: Parameter value for mobile user classification.

| Parameter | Value |
|---|---|
| Crossover probability | 0.85 |
| Mutation probability | 0.09 |
| Maximum number of iteration | 1000 |
| Weighting coefficient | $a = 0.1$, $b = 0.1$, $c = 0.1$, and $d = 0.7$ |

the mobile user classification firstly; then we will analysis the performance of the algorithm in this section.

*5.1. Experiment.* In order to verify this algorithm, we collected 1000 groups of the mobile user data and stored them in our data base. In this data set, ithas 6 feature attributes and one classification attribute; the information of these attributes is shown in Table 1.

We selected 70% of the data set as the training data and others as the test data. Then we set the parameter value as shown in Table 2.

Running the algorithm, we can get the rule as shown in Table 3.

As shown in Table 3, we can see that the most important factor which will influence the classification result is the attribute Wireless cost. If the Wireless cost is larger than 27.7 and under the public context, the user will belong to the Total service class with 84.66% accuracy in the training data and 83.25% in the test data. Then the less important factor attribute is the Free part; if Wireless cost is less than 27.7 and the Free part larger than 13.5 under the public context, the user will belong to the Plus service class with 72.68% accuracy in the training data and 70.13% in the test data; if Wireless cost is less than 27.7 and the Free part is less than 13.5 and the user under private context and the moth is less than 16, the user will belong to Basic service; this rule has 73.07% accuracy in training data and 71.64% in the test data. If Wireless cost less than 27.7 and the Free is less than 13.5 and under the private context and month is larger than 16, the user will belong to E-service; the accuracy of this rule is 64.8% in training data and 63.79% in test data. So according to above analysis, the least important is month and the most important factor is Wireless cost.

*5.2. Algorithm Analysis.* In this section, we will analysis the algorithm we proposed in this paper. We will analysis the algorithm from two parts: one is comparison on the same data, the advantage of the algorithm and the other one is analysis of the performance of the algorithm.

To verity the effects of the method we proposed in this paper, we use C4.5 algorithm and SVM method to deal with the mobile user data set. We can get 5 rules with C4.5 algorithm and 6 rules with SVM algorithm. The average accuracy of each algorithm is shown in Table 4. We can see that the accuracy of C4.5 algorithm in the mobile user training data is 68.2% and 67.9% in the test data. The accuracy of SVM algorithm in the mobile user training data is 72.5% and 70.1% in the test data. We can conclude that the accuracy of C4.5 algorithm on the mobile user is lower than that of the SVM algorithm, but the rule which is generated by the SVM will be understood hardly.

Then we put the accuracy of three algorithms in one table, to show the advantage of our algorithm that we proposed. As we can see from Table 5, the accuracy of the DT-GA algorithm that we proposed in this paper is superior to C4.5 and SVM algorithms.

Another advantage of our algorithm is simplicity; in other words, the rule which is generated by our algorithm can be understood easily. Rules generated by other algorithms will be hardly. For example, the first rule in Table 5 is

$$\begin{gathered}\text{IF } (\text{Wireless cost} \geq 27.7)\\ \text{and } (\text{Context} = \text{public context})\\ \text{THEN class} = \text{Total service.}\end{gathered} \tag{14}$$

We can understand this rule very easily. The rule by SVM

$$\begin{gathered}\text{IF } (\text{Wireless cost} \geq 27.7) \text{ and } (\text{Free part} \geq 0)\\ \text{and } (\text{Context} = \text{public}) \text{ and } (\text{month} > 0)\\ \text{THEN class} = \text{ Total service.}\end{gathered} \tag{15}$$

Table 3: Rules on mobile users generated by the algorithm.

| ID | Rule | Training set accuracy | Test set accuracy |
|---|---|---|---|
| 1 | IF (Wireless cost ≥ 27.7) and (Context = public context) THEN class = Total service | 84.66% | 83.25% |
| 2 | IF (Wireless cost ≤ 27.7) and (Free part > 13.5) and (Context = public context) THEN class = Plus service | 72.68% | 70.13% |
| 3 | IF (Wireless cost ≤ 27.7) and (Free part ≤ 13.5) and (Context = private) and (month ≤ 16) THEN class = Basic service | 73.07% | 71.64% |
| 4 | IF (Wireless cost ≤ 27.7) and (Free part ≤ 13.5) and (Context = private) and (month > 16) THEN class = E-service | 64.87% | 63.79% |

Table 4: Compare C4.5 to decision tree based on genetic algorithm on accuracy.

| Data set | Accuracy of C4.5 algorithm | Accuracy of SVM algorithm |
|---|---|---|
| Mobile user data | | |
| Training data | 68.2% | 72.5% |
| Test data | 67.9% | 70.1% |

Table 5: Comparison on accuracy.

| | Training data | Test data |
|---|---|---|
| DT-GA | 73.82% | 72.20% |
| C4.5 | 68.20% | 67.90% |
| SVM | 72.50% | 70.10% |

Table 6: Experiment parameter value.

| Parameter | Value |
|---|---|
| Crossover probability | 0.8 |
| Mutation probability | 0.1 |
| Maximum number of iteration | 500 |
| Weighting coefficient | $a = 0.1$, $b = 0.1$, $c = 0.2$, and $d = 0.6$ |

Table 7: Accuracy on Iris and Breast-cancer data.

| | Iris test data | Breast-cancer test data |
|---|---|---|
| DT-GA | 72.20% | 76.40% |
| C4.5 | 67.90% | 68.50% |
| SVM | 70.10% | 69.10% |

This rule is too long to be understood. So we can conclude that the algorithm DT-GA that we proposed in this paper is super than other algorithm not only in accuracy but also in simplicity of the rule.

We do other experiments to show the algorithm we proposed in this paper; we used Iris and breast-cancer data sets, which are two main data sets of machine learning database UCI. The numbers of samples are 150 and 286, respectively; 4 characteristic attributes and 1 class attribute are included in the Iris database, 34 characteristic attributes and 1 class attribute are included in the breast-cancer database.

Two thirds of the data sets are randomly selected as training set and the rest as test set. The class is set as default class that has the most samples. And the discretization process in advance is undesired for continuous attributes in the experiment. At the same time, there is no additional process towards default value but filling the missing value with a negative value.

In this paper, the algorithm parameter values are shown in Table 6.

Through the experiment, by comparing the accuracy of algorithm C4.5, we can get the effectiveness of our method. Firstly, C4.5 algorithm is applied to the two data sets we mentioned before; the accuracy on the training set is 68.20% and 71.90%, respectively, and the accuracy of SVM method is 72.50% and 72.60%. Then we use the algorithm to deal with the test set; the accuracy on Iris test data is 70.10% and breast-cancer test set is 69.10%. However, the accuracy of our method on the Iris training set is 73.82% and the accuracy of the breast-cancer training set is 78.50%, respectively; the accuracy of our algorithm on the Iris test set is 72.20% and the accuracy on breast-cancer test set data is 76.4%. The accuracy is as shown in Table 7 and Figure 10 is the time consumption comparison with C4.5 and SVM algorithms.

The experiment, we can conclude that the algorithm we proposed in this paper has advantage on the accuracy and simplicity of the rule.

## 6. Conclusion

In this paper we proposed a modified decision tree based on genetic algorithm; it takes advantage of genetic algorithm optimization ability. We constructed the process of this algorithm firstly, and then we do an experiment with the algorithm; through the comparison, we can conclude that the algorithm which we proposed in this paper was improved compared to normal decision tree algorithm on accuracy. At last, we applied this algorithm on mobile users, and with the algorithm we can classify the mobile user into Basic service user, E-service user, Plus service user, and Total service user

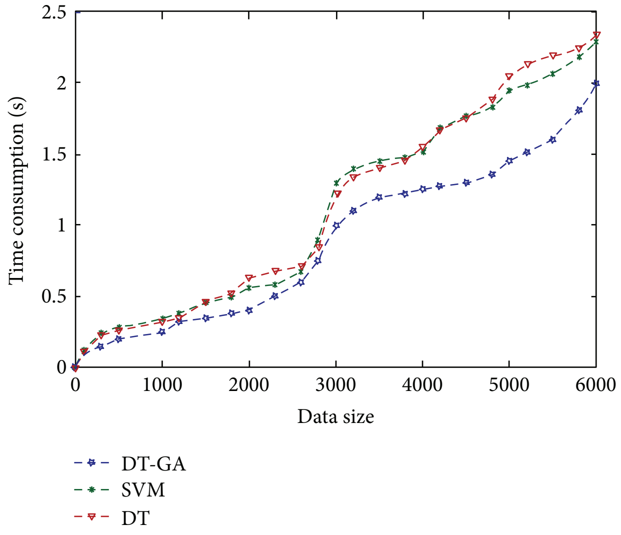


Figure 10: Time consumption comparison.

classes. Four rules with higher accuracy have been generated with the algorithm.

The further work will be on classifying the mobile user with more users' attributes and analyzing the performance of the algorithm, and we will use the algorithm to other fields if it possible, such as tourist classification, customer churn.

## Conflict of Interests

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

## Acknowledgments

This research is supported by National Natural Science Foundation of China (Grant nos. 71071140 and 71301070005), National Natural Science Foundation of Zhejiang Province (Grand no. Y1090617), and Key Innovation Team of Zhejiang Province (Grand no. 2010R50041) as well as Soft Science Key Research Project of Zhejiang Province (Grand no. 2013C25053), Modern Business Centre of Zhejiang Gong-Shang University, and Ministry of Education, Humanities and Social Sciences project (Grant no. 13YJCZH216).

## References

[1] S. H. Han, S. X. Lu, and S. C. H. Leung, "Segmentation of telecom customers based on customer value by decision tree model," *Expert Systems with Applications*, vol. 39, no. 4, pp. 3964–3973, 2012.

[2] J. Xiao, L. Xie, C. He, and X. Jiang, "Dynamic classifier ensemble model for customer classification with imbalanced class distribution," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3668–3675, 2012.

[3] N. Luo and Z. C. Mu, "Bayesian network classifier and its application in CRM," *Computer Application*, vol. 24, no. 3, pp. 79–81, 2004.

[4] W.-Y. Chiang, "Applying a new model of customer value on international air passengers' market in Taiwan," *International Journal of Tourism Research*, vol. 14, no. 1, pp. 116–123, 2012.

[5] M. -C. Zheng, *A Comparative Analysis of Classifying Algorithms in Data Mining Technology*, Lanzhou University of finance and Economics, 2007.

[6] C.-E. Zhang, *Banks' Personal Client Segmentation Research Based on Decision Tree*, North China Electric Power University, 2007.

[7] Y.-J. Chen, "Design and implementation of a hibernal tree classification system based on bayes," *Computer Applications and Software*, vol. 26, no. 6, pp. 178–180, 2009.

[8] Y. Zhou, A.-F. Zhu, L. Zhou, and X. Qian, "Sample data selection method for neural network classifiers," *Journal oF Huazhong University of Science and Technology*, vol. 40, no. 6, pp. 39–43, 2012.

[9] Z. Shu, *The Research of Fingerprint Classification Based on Genetic Algorithm*, Huazhong University of Science and Technology, 2006.

[10] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, New York, NY, USA, 1993.

[12] G. Kass, "An exploratory technique for investigating large quantities of categorical data," *Applied Statistics*, vol. 29, no. 2, pp. 119–127, 1980.

[13] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Chapman and Hall/CRC, New York, NY, USA, 1984.

[14] J. R. Quinlan, "Bagging, boosting, and C4.5," in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 725–730, Portland, Oregon, August 1996.

[15] Y. Freund and R. E. SchPaire, "A decision theoretic generalization of on line learning and an application to boosting," *Computational Learning Theory*, vol. 904, pp. 23–37, 1997.

[16] S.-L. Pang and J.-Z. Gong, "C5.0 classification algorithm and its application on individual credit score for banks," *System Engineering Theory and Practice*, vol. 29, no. 12, pp. 94–104, 2009 (Chinese).

[17] J. Yang and Z. Fei, "Broadcasting with prediction and selective forwarding in vehicular networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 309041, 9 pages, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 941532, 21 pages
http://dx.doi.org/10.1155/2014/941532

*Research Article*

# Hierarchical Artificial Bee Colony Algorithm for RFID Network Planning Optimization

**Lianbo Ma,[1,2] Hanning Chen,[1] Kunyuan Hu,[1] and Yunlong Zhu[1]**

[1] *Department of Information Service & Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, Faculty Office VII, Nanta Street No. 114, Dongling District, Shenyang 110016, China*
[2] *University of Chinese Academy of Sciences, Beijing 100039, China*

Correspondence should be addressed to Lianbo Ma; malianbo521@gmail.com and Hanning Chen; chenhanning@sia.cn

Received 22 October 2013; Accepted 4 December 2013; Published 23 January 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

This paper presents a novel optimization algorithm, namely, hierarchical artificial bee colony optimization, called HABC, to tackle the radio frequency identification network planning (RNP) problem. In the proposed multilevel model, the higher-level species can be aggregated by the subpopulations from lower level. In the bottom level, each subpopulation employing the canonical ABC method searches the part-dimensional optimum in parallel, which can be constructed into a complete solution for the upper level. At the same time, the comprehensive learning method with crossover and mutation operators is applied to enhance the global search ability between species. Experiments are conducted on a set of 10 benchmark optimization problems. The results demonstrate that the proposed HABC obtains remarkable performance on most chosen benchmark functions when compared to several successful swarm intelligence and evolutionary algorithms. Then HABC is used for solving the real-world RNP problem on two instances with different scales. Simulation results show that the proposed algorithm is superior for solving RNP, in terms of optimization accuracy and computation robustness.

## 1. Introduction

In recent years, radio frequency identification (RFID) technology as a new inventory tracking technology has great promise for diversified use in many industries with numerous practical applications. Much great potential has been realized and many are being explored. In many real-world RIFD applications, such as production, logistics, supply chain management, and asset tracking, a sufficient number of readers are deployed in order to provide complete coverage of all the tags in the given area [1, 2]. Specially, over the last ten years, RFID is used to build up an "Internet of Things" (IoT), a network that connects physical things to the Internet, making it possible to access remote sensor data and to control the physical world from a distance [3]. This brings some questions in the deployment of an RFID network for the operation and management of the large-scale Internet of Things applications, such as optimal tag coverage, quality of service (QoS), and cost efficiency [4].

Due to the limited recognition range of single reader, many readers and tags need to be deployed according to some arrangement to construct the RFID system in the scenario area. This results in some necessary questions to be considered in the case of avoiding reader collision, such as (1) how many readers are needed; (2) where should the readers be placed; (3) what is the efficient parameter setting for each reader [5, 6]. In addition, considering cost-efficient for RFID system, the network should meet the items with minimum number of readers and maximum tags coverage. Thus, the RFID network planning problem (RNP) is a difficult NP problem [7, 8]. In general, we defined that the RNP aims to optimize a set of objectives (tags coverage, load balance, economic efficiency, interference between readers, etc.), by adjusting the control variables (the coordinates of the readers, the number of the readers, the antenna parameters, etc.) of the system [8].

In the past two decades, evolutionary computation (EC) and swarm intelligence (SI) techniques for solving RNP problem have gained increasing attention, such as genetic algorithms (GA) [9, 10], evolutionary strategy (ES) [11], differential evolution (DE) [12], particle swarm optimization (PSO) algorithms [6, 11, 13], and bacterial foraging algorithms (BFA) [14, 15]. Specially, in [6] we present a multispecies particle swarm optimization model for solving RNP problem, achieving a significant positioning accuracy. It is noted that some scholars propose other methods to dispose similar problems, such as bipartite graph [16, 17]. However, with the increasing number of the deployed readers and tags in the large-scale RFID deployment environment, the degree of complexity for solving the RNP optimization increases exponentially. The previous methods to solve the RNP optimization are incompetent for being prone to premature convergence [13].

A natural approach to tackle high-dimensional optimization problems is to adopt the cooperative coevolution based on divide-and-conquer strategy. An early work on a cooperative co-evolutionary algorithm (CCEA) by [18] provides a promising approach for decomposing a high-dimensional problem. Recent studies [19–23] by taking improved decomposition strategy into PSO algorithm. Inspired by these recent works, we propose a novel hierarchical coevolving scheme, extending the canonical artificial bee colony (ABC) algorithm framework from nonhierarchy to hierarchy, called hierarchical artificial bee colony algorithm (HABC). Our HABC model is inherently different from others in the following aspects.

Firstly, the cooperative coevolving approach based on divide-and-conquer strategy with random grouping technology is adopted into HABC, which enhances the local search ability (exploitation). Under this method, the high-dimensional vectors can be decomposed into smaller subcomponents which are assigned to the lower hierarchy. This method enhances the local searching ability.

Secondly, the traditional evolution operators such as crossover and mutation are applied to interaction of multispecies instead of single species enhancing the information exchange between populations. Under this new development, the neighbor bees with higher fitness can be chosen for crossover and mutation, which effectively enhances the global search and convergence to the global best solution as the dimension increases. This maintains population diversity and enhances the global search ability (exploration).

By incorporating this new degree of complexity, HABC can accommodate considerable potential for solving more complex problems. Here we provide some initial insights into this potential by evaluating HABC on both mathematical benchmark functions and a real-world RNP case, which focuses on minimizing four specific objective functions of a 30-reader RFID network and a 50-reader RFID network. The simulation results, which are compared to other state-of-the-art methods, show the superiority of the proposed algorithm.

The rest of the paper is organized as follows. In Section 2, RFID system models and the RNP problem definitions are presented. Section 3 first gives a review of the canonical ABC algorithm and then proposes the novel HABC algorithm. Section 4 tests the algorithm on the ten benchmarks and illustrates the results. Section 5 describes the implementation of the proposed approach based on HABC on two instances, Cd100 and Rd100, and the results of simulation are analyzed. Finally, Section 6 outlines the conclusions.

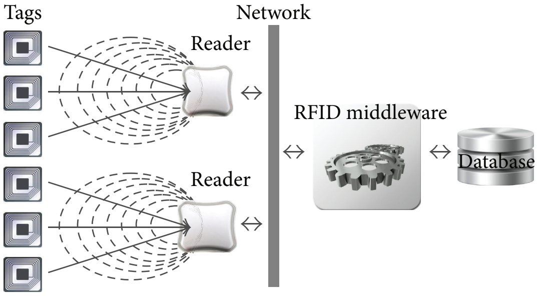


Figure 1: RFID system.

## 2. Problem Formulation on RNP

An RFID system consists of four types of important components (see Figure 1): (1) RFID tags, each placed on an object and consists of a microchip and an embedded antenna containing a unique identity, which is called Electronic Product Code (EPC); (2) RFID readers, each has more than one antenna and is responsible to send and receive data to and from the tag via radio frequency waves; (3) RFID middleware, which manages readers and filters and formats the RFID raw tag data; (4) RFID database, which records RFID raw tag data that contain information such as reading time, location, and tag EPC. In this section, a mathematical optimization model for the RNP problem based on RFID middleware is proposed.

The model is constructed from several different aspects. The deployment region of hotspots is supposed to be a two-dimension square domain. The tags here are passive and are based on the Class-1 Generation 2 UHF standard specification [6, 9, 12]. It means that they can only be powered by radio frequency energy from readers. The proposed RNP model aims to improve the QoS of RFID networks by optimizing the objects including coverage, interference, load balance, and aggregate efficiency via regulating the parameters of RFID networks, including the number, location, and radiated power of readers. Generally the problem is formulated as follows.

### 2.1. Optimal Tag Coverage ($f_c$).

The first objective function represents the level of coverage, which is the most important in an RFID system. In this paper, if the radio signal received at a tag is higher than the threshold $\delta = -10\,\text{dBm}$, the communication between reader and tag can be established. Then the function is formulated as the sum of the difference between the desired power level $\delta$ and the actual received power $P_i^j$ of each tag $i$ in the interrogation region of reader $j$:

$$\text{Min } f_c = \sum_{i \in \text{TS}} \sum_{j \in \text{RS}_i} \left(P_i^j - \delta\right), \tag{1}$$

where TS and RS are the tag and reader set that are deployed in the working area respectively, and $\mathrm{RS}_i$ represents the set of readers which has the tag $i$ in its interrogation region. This object function ensures that the received power $P_i^j$ at the tag $i$ from the reader $j$ in $\mathrm{RS}_i$, which is mainly determined by the relative distance and radiated power of the reader $j$, is higher than the threshold $\delta$, which guarantees that the tag is activated. That is, by regulating the locations and radiated power of the readers, the optimization algorithm should locate the RFID readers close to the regions where the desired coverage level is higher, while the areas requiring lower coverage are taken into account by the proper radiated power increases of the readers.

*2.2. Reader Interference* ($f_i$). Reader collision mainly occurs in a dense reader environment, where several readers try to interrogate tags at the same time in the same area. This results in an unacceptable level of misreads. The main feature of our approach is that the interference is not solved by traditional ways, such as frequency assignment [24] and reader scheduling [6], but in a more precautionary way. This objective function is formulated as

$$\text{Min } f_i = \sum_{k\in \mathrm{RS}} \sum_{i\in \mathrm{TS}_k} \left(\delta - \left(P_i^k - \sum_{j\in \mathrm{RS}_i}^{j\neq k} P_i^j\right)\right), \tag{2}$$

where $\mathrm{TS}_k$ is the tag set in the interrogation region of reader $k$. For each tag $i$, this objective considers all the readers except the best one as interfering sources. That is, by changing reader positions and powers according to this functional the algorithm tries to locate the readers far from each other to reduce the interference.

*2.3. Economic Efficiency* ($f_e$). This aspect could be approached from various points of view. For example, due to the stochastic noise, multipath effect, and attenuation in the propagation channel, readers should be located closely to the center of tags in the hotspots. From this perspective, this objective can be reached by weighing the distances of each center of tag clusters from its best served reader. Here we employ $K$-means clustering algorithm to find the tag cluster. It can be defined below as follows:

$$\text{Min } f_e = \sum_{k\in \mathrm{RS}} \mathrm{dist}\left(R_k, \theta_k\right), \tag{3}$$

where dist() is the distance between the $k$th reader and the $k$th tag center and $\theta_k$ and $R_k$ are the position of $k$th cluster center and its best served reader, respectively. In this way the algorithm tries to reduce the distance from the readers to the elements with high tag densities.

*2.4. Load Balance* ($f_b$). A network with a homogeneous distribution of reader cost can give a better performance than an unbalanced configuration [25]. Thus, in large-scale RFID system, the set of tags to be monitored needs to be properly balanced among all readers. This objective function is formulated as

$$\text{Min } f_l = \prod_{k\in \mathrm{RS}} \left(\frac{C_k^{\max}}{C_k}\right), \tag{4}$$

where $C_k$ is the assigned tags number to reader $k$ and $C_k^{\max}$ is the maximum number of tags which can be read by the reader $k$ in unit time. It should be noticed that the $C_k^{\max}$ takes different values according to the different types of readers used in the network. This object aims to minimize the variance of load conditions by changing the locations and radiated power of readers.

*2.5. Combined Measure* ($f_m$). In this paper, the overall optimal solution for RNP is represented by a linear combination of the four objective functions:

$$\begin{gathered} \text{Minimize } f_c = \sum_{i=1}^{4} \frac{w_i f_i}{f_{i\max}}; \\ w_1 + w_2 + w_3 + w_4 = 1, \quad w_i > 0, \end{gathered} \tag{5}$$

where $f_i$ is the objective function for the $i$th requirement normalized to its maximum value $f_{i\max}$; the normalization is necessary because these four objectives represent nonhomogeneous quantities and are very different in values.

*2.6. Objective Constraint.* All the tags in working area must be covered by a reader. This constraint can be formally expressed by the following formula:

$$\begin{gathered} \text{s.t. } P_i^j - \delta \geq 0 \quad \forall i \in \mathrm{TS},\ j \in \mathrm{RS}_i, \\ \sum_{k\in \mathrm{RS}} \lambda_i^k \geq 1 \quad \forall i \in \mathrm{TS}, \end{gathered} \tag{6}$$

where $\lambda_i^k$ is a binary variable in which $\lambda_i^k = 1$ if the reader $k \in \mathrm{RS}_i$; otherwise $\lambda_i^k = 0$. So this constraint can maintain the power efficiency of network and ensure a complete coverage deployment.

## 3. Hierarchical Artificial Bee Colony Algorithm

*3.1. Canonical Artificial Bee Algorithm.* The ABC algorithm is a relatively new SI algorithm by simulating the foraging behaviors of honey bee swarm, initially proposed by Karaboga and further developed by Basturk and Akay [26–28]. In ABC, the colony of artificial bees contains three groups of individuals, namely, the employed, onlookers and scouts bees. Employed bees exploit the specific food sources and give the quality information to the onlooker bees. Onlooker bees receive information about the food sources and choose a food source to exploit depending on the quality information. The employed bee whose food source has been abandoned becomes a scout and starts to search for a new food source. The fundamental mathematic representations are listed as follows.

```
Original ABC algorithm
(1) Initialization.
Initialize the food sources and evaluate the nectar amount (fitness) of food sources;
Send the employed bees to the current food source;
Iteration = 0;
(2) Do while (the termination conditions are not met)
(3)     /*Employed Bees’ Phase*/
      for (each employed bee)
find a new food source in its neighborhood following (8);
Evaluate the fitness of the new food source, Apply greedy selection;
end for
(4)     Calculate the probability $P$ for each food source;
(5)     /*Onlooker Bees’ Phase*/
    for (each onlooker bee)
Send onlooker bees to food sources depending on $P$;
find a new food source in its neighborhood following (8);
Evaluate the fitness of the new food source, Apply greedy selection;
end for
(6)     /*Scout Bees’ Phase*/
        if (any employed bee becomes scout bee)
Send the scout bee to a randomly produced food source;
end if
(7)    Memorize the best solution achieved so far
        Iteration = Iteration + 1;
end while
(8) Output the best solution achieved
```

Algorithm 1: Pseudocode of the original ABC algorithm.

In initialization phase, the algorithm generates a group of food sources that correspond to the solutions in the search space. The food sources are produced randomly within the range of the boundaries of the variables. Consider

$$x_{i,j} = x_j^{\min} + \text{rand}\,(0,1)\left(x_j^{\max} - x_j^{\min}\right), \tag{7}$$

where $i = 1, 2, \ldots, \text{SN}$ and $j = 1, 2, \ldots, D$. SN is the number of food sources and equals to half of the colony size. $D$ is the dimension of the problem, representing the number of parameters to be optimized. $x_j^{\min}$ and $x_j^{\max}$ are lower and upper bounds of the $j$th parameter, respectively. Additional, counters which store the numbers of trials of each bee are set to 0 in this phase.

In the employed bees’ phase, each employed bee is sent to the food source in its memory and finds a neighboring food source. The neighboring food source is produced according to (8) as follows:

$$v_{i,j} = x_{i,j} + \varphi\left(x_{i,j} - x_{k,j}\right), \tag{8}$$

where $k$ is a randomly selected food source different from neighbor $i$. $j$ is a randomly selected dimension. $\phi$ is a random number which is uniformly distributed in range $[-1, 1]$. The new food source $v$ is determined by changing one dimension on $x$. If the value in this dimension produced by this operation exceeds its predetermined boundaries, it will set to be the boundaries.

The new food source is then evaluated. A greedy selection is applied on the original food source and the new one. The better one will be kept in the memory. The trials counter of this food will be reset to zero if the food source is improved; otherwise, its value will be incremented by one.

In the onlooker bees’ phase, the onlookers receive the information of the food sources shared by employed bees. Then they will choose a food source to exploit depending on a probability related to the nectar amount of the food source (fitness values of the solution). That is to say, there may be more than one onlooker bee choosing the same food source if the source has a higher fitness. The probability is calculated according to (9) as follows:

$$P_i = \frac{\text{fitness}_i}{\sum_{j=1}^{\text{SN}} \text{fitness}_j}. \tag{9}$$

After food sources have been chosen, each onlooker bee finds a new food source in its neighborhood following (8), just like the employed bee does. A greedy selection is also applied on the new and original food sources.

In scout bees’ phase, if a food source has not been improved for a predetermined cycle, which is a control parameter called “limit,” the food source is abandoned and the bee becomes a scout bee. A new food source will be produced randomly in the search space using (7), as in the case of initialization phase.

The employed, onlooker, and scout bees’ phase will recycle until the termination condition is met. The best food source which presents the best solution is then outputted. The pseudo-code of original ABC algorithm is illustrated in Algorithm 1.

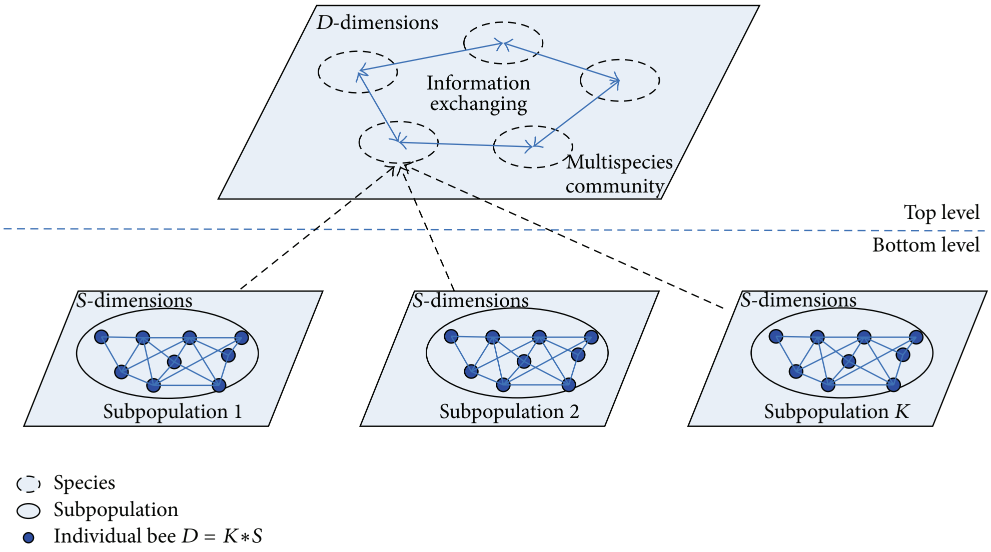


Figure 2: Hierarchical optimization model.

*3.2. The Hierarchical Artificial Bee Colony Algorithm.* The HABC integrates a two-level hierarchical co-evolution scheme inspired by the concept and main ideas of multi-population co-evolution strategy and cross and mutation operations. The flowchart of the HABC is shown in Figure 1. It includes four important strategy approaches: variables decomposing approach, random grouping of variables, background vector calculating approach, and cross and mutation operation, which is presented as follows.

*3.2.1. Hierarchical Multipopulation Optimization Model.* As described in Section 3.1, we can see that the new food source is produced by a perturbation coming from a random single dimension in a randomly chosen bee. This causes that an individual may have discovered some good dimensions, while the other individuals that follow this bee are likely to choose worse vectors in $D$ dimensions and abandon the good ones. On the other hand, when solving complex problems, the canonical ABC algorithm based on single population suffers from the following drawback: as a population evolves, all individuals suffer premature convergence to the local optimum in the first generations. This leads to low population diversity and adaptation stagnation in successive generations.

Hence, the HABC contains two levels, namely, the bottom level and top level, to balance exploring and exploiting ability. In Figure 2, in the bottom level, with the variables decomposing strategy, each subpopulation employs the canonical ABC method to search the part-dimensional optimum in parallel. That is, in each iteration, $K$ subpopulations in the bottom level generate $K$ best solutions, which are constructed into a complete solution species that update to the top level. In the top level, the multispecies community adopts the information exchange mechanism based on crossover operator, by which each species can learn from its neighborhoods in a specific topology. The vectors decomposing strategy and information exchange (i.e., crossover operator) can be described in detail as follows.

*3.2.2. Variables Decomposing Approach.* The purpose of this approach is to obtain finer local search in single dimensions inspired by the divide-and-conquer approach. Notice that two aspects must be analyzed: (1) how to decompose the whole solution vector, and (2) how to calculate the fitness of each individual of each subpopulation. The detailed procedure solving those is presented as follows.

*Step 1.* The simplest grouping method is permitting a $D$-dimensional vector to be split into $K$ subcomponents, each corresponding to a subpopulation of $s$-dimensions, with $M$ individuals (where $N = K * s$). The $j$th subpopulation is denoted as $P_j$, $j \in [1, \dots, K]$.

*Step 2.* Construct complete evolving solution $G_{\text{best}}$ which is the concatenation of the best subcomponents' solutions $P_j$ by the following:

$$G_{\text{best}} = \left(P_1 \cdot g, P_2 \cdot g, P_j \cdot g, \dots, P_K \cdot g\right), \tag{10}$$

where $P_j \cdot g$ represents the personal best solution of the $j$th subpopulation.

*Step 3.* For each component $P_j$, $j \in [1, \dots, K]$, do the following:

(a) *at employed bees' phase*, for each individual $X_i$, $i \in [1, \dots, M]$; replace the $i$th component of the $G_{\text{best}}$ by using the $i$th component of individual $X_i$; calculate the new solution fitness: $f(\text{new}\, G_{\text{best}}(P_1 \cdot g, P_2 \cdot g, X_i, \dots, P_k \cdot g))$. If $f(\text{new}\, G_{\text{best}}) < f(G_{\text{best}})$, then $G_{\text{best}}$ is replaced by new $G_{\text{best}}$.

(b) update $X_i$ positions by using (8);

(c) at onlooker Bees' Phase, repeat (a)-(b).

*Step 4.* Memorize the best solution achieved so far; compare the best solution with $G_{\text{best}}$ and memorize the better one.

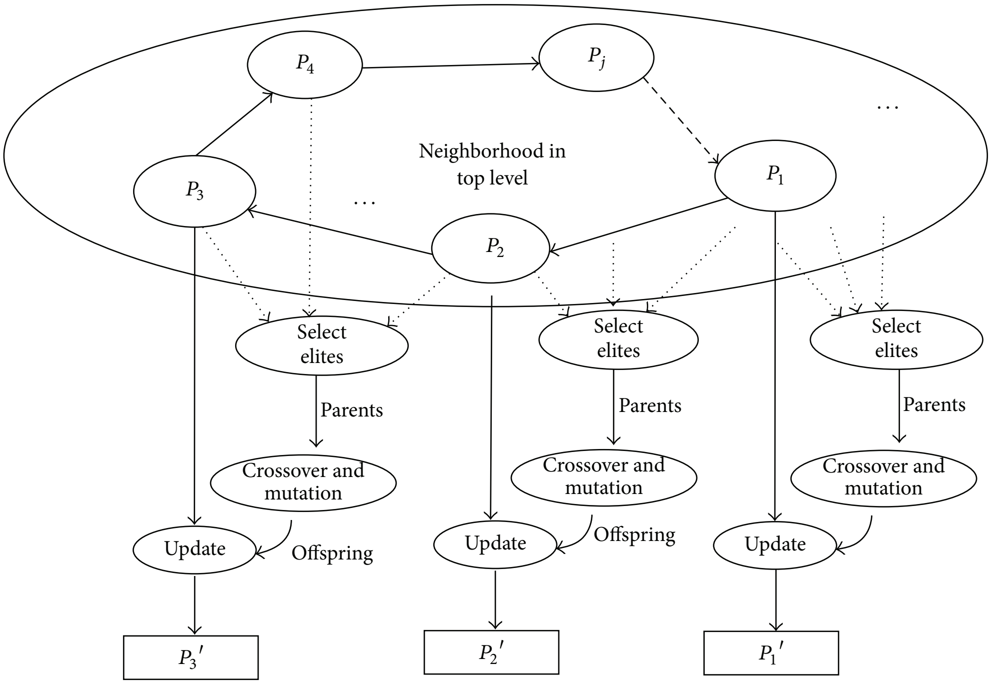


Figure 3: The information exchange mechanism based on crossover operator.

Under this method, high-dimensional objective function vectors can be decomposed into smaller subcomponents, which are evolving separately. This multipopulations parallel processing approach enhances the local searching ability.

*Random Grouping of Variables*. To increase the probability of two interacting variables allocated to the same subcomponent, without assuming any prior knowledge of the problem, according to the random grouping of variables proposed by [20, 21], we adopt the similar random grouping scheme by dynamically changing group size. For example, for a problem of 100 dimensions, we can define that

$$G = \{2, 5, 10, 20, 100\},$$
$$K \subset G. \tag{11}$$

Here, if we randomly decompose the $D$-dimensional object vector into $K$ subcomponents in each iteration (i.e., we construct each of the $K$ subcomponents by randomly selecting $S$-dimensions from the $D$-dimensional object vector), the probability of placing two interacting variables into the same subcomponent becomes higher over an increasing number of iterations.

### *3.2.3. The Information Exchange Mechanism Based on Crossover Operator between Multispecies.*

In the top level, we adopt crossover operator with a specific topology to enhance the information exchange between species, in which each species $P_j$ can learn from its symbiotic partner in the neighborhood. The key operations of this crossover procedure are described in Figure 3.

*Step 1* (select elites to the best-performing list (BPL)). First, a set of competent individuals from current species $P_j$'s neighborhood (i.e., ring topology) are selected to construct the best-performing list (BPL) with higher fitness has larger probability to be selected. The size of BPL is equal to the number of current species $P_j$. These individuals of BPL are regarded as elites. The selection operation tries to mimic the maturing phenomenon in nature, where the generated offspring will become more suitable to the environment by using these elites as parents.

*Step 2* (crossover and mutation between species). To produce well-performing individuals, parents are selected from the BPL's elites only for the crossover operation. To select parents effectively, the tournament selection scheme is used, in which two enhanced elites are selected randomly, and their fitness values are compared to select the elites, and the one with better fitness value is regarded as parent. Then another parent is selected in the same way. Two offsprings are created by performing crossover on the selected parents. This paper adopts the arithmetic crossover method: the offspring is produced by

$$S_{\text{new}} = \text{rand}(0,1) \times \text{parent } 1 + \text{rand}(0,1) \times \text{parent } 2, \tag{12}$$

where $S_{\text{new}}$ is the newly produced offspring and parent 1 and parent 2 are randomly selected from BPL.

*Step 3* (update with greedy selection strategy). Not all current species are replaced by the elites from BPL; we set a selecting rate CR to determine the replaced individuals. Assuming that species size of $P_j$ is $M$, then the replaced individuals number is $M * \text{CR}$. For the selected individual $S_j$, the newly produced

Table 1: Parameters setting for all algorithms.

| Type | HABC | ABC | CPSO | PSO | CCEA | EGA | CMA-ES |
|---|---|---|---|---|---|---|---|
| $N$ | 10 | NA | NA | NA | NA | NA | NA |
| $M$ | 5 | 50 | 5 | 50 | 5 | 50 | 50 |
| $K$ | $\{2, 5, 10, 50\}$ | NA | 50 | NA | 50 | NA | NA |
| CR | 1 | NA | NA | NA | NA | NA | NA |
| $\chi$ | NA | NA | 0.729 | 0.729 | NA | NA | NA |
| $c1$ | NA | NA | 2.05 | 2 | NA | NA | NA |
| $c2$ | NA | NA | 2.05 | 2 | NA | NA | NA |
| $\mu/c_\sigma/d_\sigma/c_c/c_{\text{cov}}$ | NA | NA | NA | NA | NA | NA | 12/0.1/20/0.12/0.08 |
| Crossover rate | NA | NA | NA | NA | 0.6 | 0.8 | NA |
| Mutation rate | NA | NA | NA | NA | 0.02 | 0.01 | NA |

offspring $S_{\text{new}}$ is then compared with $S_j$, applying a greedy selection mechanism, in which the better one is remained. We can choose four selecting approaches: selecting the best individuals (i.e., $M * \text{CR}$ individuals), a medium level of individuals, the worst individuals, and random individuals. Hence, there are several HABC variants according to different selecting approach. Here, we choose the simplest approach (i.e., selecting the worst individuals) to be replaced.

## 4. Benchmark Test

In the experimental studies, according to the no free lunch (NFL) theorem [29], a set of 10 benchmark functions, which are listed in Appendix A, are employed to fully evaluate the performance of the HABC algorithm without a biased conclusion towards some chosen problems. In order to compare the different algorithms fairly, we decide to use the number of function evaluations (FEs) as a time measure substituting the number of iterations due to the reason that the algorithms do differing amounts of work in their inner loops.

*4.1. Experimental Settings.* Experiments are conducted with six variants of HABC according to the different CR values. The proposed HABC is compared with six successful EA and SI algorithms: artificial bee colony algorithm (ABC) [26], cooperative co-evolutionary algorithm (CCEA) [18], canonical PSO with constriction factor (PSO) [30], cooperative PSO (CPSO) [19], genetic algorithm with elitism (EGA) [31], and covariance matrix adaptation evolution strategy (CMA-ES) [32].

ABC is a recently developed SI paradigm simulating foraging behavior of bees [26]. CCEA is the earliest cooperative coevolutionary algorithm which applied the divide-and-conquer approach by Potter and de Jong [18]. CPSO is a cooperative PSO model, cooperatively coevolving multiple PSO subpopulations [19]. EGA is the classical genetic algorithm with elitist selection scheme [31]; the underlying idea of CMA-ES is to gather information about successful search steps and to use that information to modify the covariance matrix of the mutation distribution in a goal directed, derandomized fashion [32].

In all experiments in this section, the values of the common parameters used in each algorithm such as population size and total generation number are chosen to be the same. Population size is set as 50 and the maximum evaluation number is set as 100000. For the fifteen continuous testing functions used in this paper, the dimensions are all set as $100D$.

All the control parameters for the EA and SI algorithms are set to be default of their original literatures: initialization conditions of CMA-ES are the same as in [32], and the number of offspring candidate solutions generated per time step is $\lambda = 4\mu$; the limit parameter of ABC is set to be $\text{SN} \times D$, where $D$ is the dimension of the problem and SN is the number of employed bees. The split factor for CCEA and CPSO is equal to the dimensions [18, 19]. For canonical PSO and CPSO, the learning rates $c_1$ and $c_2$ are both set as 2.05 and the constriction factor $\chi = 0.729$. For EGA, intermediate crossover rate of 0.8, Gaussian mutation rate of 0.01, and the global elite operation with a rate of 0.06 are adopted [31]. The parameter setting for all algorithms is listed in Table 1. For the proposed HABC, the species number $N$, split factor $K$ (i.e., the subpopulation number), and the selection rate CR should be tuned firstly in the next section.

*4.2. Sensitivity in relation to Parameters of HABC*

*4.2.1. Effects of Species Number $N$.* The number of species of HABC in top level needs to be tuned. Three benchmarks-Sphere, Rosenbrock, and Schwefel are used to investigate the impact of this parameter. The selection rate can be set as $\text{CR} = 1$, and the involved benchmark functions (i.e., Sphere, Rosenbrock, and Schwefel) are run 50 times. From Figure 5, we can observe that when $N$ increased, we obtained faster convergence velocity and better results on all test functions. However, it can be observed that the performance improvement is not evident when $N > 10$ for most test functions. Thus, in our experiments, the species number $N$ for HABC is set at 10 for all test functions.

*4.2.2. Choices of CR.* The basic benchmark functions ($f_1$–$f_5$) are adopted to evaluate the performance of HABC variants with different CRs. Form Table 2, we can find that HABC variant with CR equal to 1 performed best on four functions

Table 2: Performance of HABC on $100D$ benchmark functions $f_1$–$f_5$ with different CRs. In bold are the best results. Note that values indicated in brackets denoted to function evaluations (FEs) are used to obtain the optimal value.

| CR | | 0.05 | 0.1 | 0.2 | 0.4 | 0.6 | 1 |
|---|---|---|---|---|---|---|---|
| $f_1$ | Mean | 1.69e − 109 | 3.74e − 215 | 4.25e − 290 | 0 (64800) | 0 (56000) | **0 (36000)** |
| | Std | 3.79e − 109 | 1.79e − 201 | 3.19e − 200 | 4.79e − 210 | 6.59e − 200 | 3.06e − 200 |
| $f_2$ | Mean | 2.17e − 004 | 1.16e − 006 | 1.08e − 005 | 5.50e − 005 | 4.26e − 006 | **4.35e − 007** |
| | Std | 4.07e − 004 | 2.1e − 006 | 8.15e − 006 | 12.30 | 6.38e − 006 | **5.69e − 007** |
| $f_3$ | Mean | 2.36e − 101 | 2.99e − 194 | 1.71e − 267 | 0 (69700) | 0 (49800) | **0 (38000)** |
| | Std | 4.74e − 101 | 8.79e − 109 | 3.01e − 109 | 9.89e − 109 | 7.87e − 100 | 1.61e − 110 |
| $f_4$ | Mean | **1.93e − 028** | 4.46e − 028 | 4.97e − 028 | 5.14e − 026 | 7.62e − 25 | 7.87e − 022 |
| | Std | **2.03e − 028** | 8.01e − 028 | 3.44e − 028 | 5.77e − 026 | 5.55e − 025 | 1.34e − 021 |
| $f_5$ | Mean | 0 (20000) | 0 (13500) | 0 (7900) | 0 (6860) | 0 (4400) | **0 (3100)** |
| | Std | 1.21e − 200 | 1.79e − 209 | 3.71e − 209 | 4.75e − 209 | 6.01e − 209 | 1.21e − 209 |

Table 3: Performance of HABC on $100D$ benchmark functions $f_1$–$f_4$ with the different grouping number $K$. In bold are the best results. Note that values indicated in brackets denoted to function evaluations (FEs) are used to obtain the optimal value.

| Func. | | $K \subset S$ | $K = 5$ | $K = 10$ | $K = 50$ | $K = 100$ |
|---|---|---|---|---|---|---|
| $f_1$ | Mean | **0 (91000)** | **0 (45000)** | 2.00e − 080 | 3.70e − 043 | 6.56e − 042 |
| | Std | **0** | **0** | 1.03e − 001 | 6.31e − 043 | 1.27e − 041 |
| $f_2$ | Mean | 1.10e − 014 | 1.47 | 1.56 | 6.50e − 001 | **1.61e − 022** |
| | Std | 1.37e − 014 | 3.12e − 001 | 2.47e − 001 | 1.13 | **1.38e − 022** |
| $f_3$ | Mean | **0** | **0** | 7.25e − 014 | 5.39e − 013 | 8.22e − 013 |
| | Std | **0** | **0** | 1.45e − 013 | 2.32e − 013 | 1.33e − 013 |
| $f_4$ | Mean | **1.70e − 003** | 7.69e − 002 | 4.45e − 002 | 3.23e + 002 | 1.99e + 002 |
| | Std | **1.10e − 003** | 1.11e − 001 | 2.71e − 002 | 5.37e + 002 | 3.94e + 002 |

among all five functions, while CR equal to 0.05 got the best result on one function. According to the results with different CRs, we chose CR equal to 1 as an optimal value for the next experiments.

*4.2.3. Effects of Dynamically Changing Group Size $K$.* Obviously, the choice of value for split factor $K$ (i.e., subpopulation number) had a significant impact on the performance of the proposed algorithm. In order to vary $K$ during a run, we defined $S = \{2, 5, 10, 50, 100\}$ for $100D$ function optimization and the split factor $K$ is chosen uniformly at random from a set $S$. Then, the variant of HABC with dynamically changing $K$ is compared with that with fixed split number on four benchmark functions for 50 sample runs. From the results listed in Table 3, we can observe that the performance is sensitive to the predefined $K$ value. HABC, using a dynamically changing $K$ value, consistently gave a better performance than the other variants except $f_2$. Moreover, in most real-world problems, we do not have any prior knowledge about the optimal $s$ value, so the random grouping scheme can be a suitable solution.

*4.3. Comparing HABC with Other State-of-the-Art Algorithms on CEC Benchmark Functions.* To verify the effectiveness of the proposed algorithm, CEC2005 functions ($f_6$–$f_{10}$) are adopted to evaluate the performance of all algorithms. According to Section 4.2, we ensure the following optimal parameter setting of HABC: CR $= 1$, $N = 10$, and $K \subset S$, in comparison with CCEA, CPSO, CMA-ES, ABC, PSO, and EGA algorithms. Table 4 showed the experimental results (i.e., the mean and standard deviations of the error values ($f(x) - f(x^*)$) values found in 50 sample runs) for each algorithm on $f_6$–$f_{10}$. Figure 6 shows the search progress of the average values for all algorithms.

The mentioned five shifted and rotated functions $f_6$–$f_{10}$ are regarded as the most difficult functions to optimize. HABC outperformed CMA-ES on three out of all the five functions, except $f_7$ and $f_{10}$. HABC can find the global optimum for $f_6$ within 10000 FEs, because the adopted variables decomposing approach can enhance the local search, which is a key contributing factor to handle high-dimensional problem. On the other hand, CMA-ES converges extremely fast. However, it either converged very well or tended to become stagnant very quickly, especially on the multimodal functions ($f_9$,$f_{10}$). From the rank values presented in Table 4, the performance of all algorithms tested here is ordered as HABC > CMA-ES > ABC > CCEA > CPSO > PSO > EGA.

## 5. RFID Network Planning Based-HABC Algorithm

In this section, the details of proposed approach to solve the RNP problem are described.

*5.1. Solution Representation of RNP Problem.* In this work, the task of RFID network planning is to deploy several RFID

Table 4: Performance of all algorithms on $100D$ benchmark functions $f_6$–$f_{10}$. In bold are the best results.

| Func. | | HABC | ABC | CPSO | CMA-ES | CCEA | PSO | EGA |
|---|---|---|---|---|---|---|---|---|
| | Mean | **0** | $1.13e-013$ | $3.37e+002$ | $5.68e-014$ | $2.27e-012$ | $2.06e+001$ | $7.12e+001$ |
| $f_6$ | Std | $1.12e-021$ | $4.92e-014$ | $5.84e+002$ | $4.92e-014$ | $1.58e-012$ | $4.61e+001$ | $1.66e+001$ |
| | Rank | **1** | 3 | 7 | 2 | 4 | 5 | 6 |
| | Mean | $3.91e+001$ | $2.10e+003$ | $6.14e+003$ | **0** | $4.50e+003$ | $1.44e+002$ | $8.93e+003$ |
| $f_7$ | Std | $4.96e+001$ | $1.52e+002$ | $1.17e+004$ | $\mathbf{5.68e-014}$ | $1.32e+003$ | $1.10e+002$ | $5.18e+003$ |
| | Rank | 2 | 4 | 6 | **1** | 5 | 3 | 7 |
| | Mean | 1.39 | $1.21e+001$ | $1.33e+006$ | 1.59 | 6.74 | $3.20e+001$ | $2.66e+004$ |
| $f_8$ | Std | $\mathbf{8.90e-001}$ | 7.09 | $1.21e+006$ | 2.18 | 4.47 | $3.31e+001$ | $2.11e+004$ |
| | Rank | **1** | 4 | 7 | 2 | 3 | 5 | 6 |
| | Mean | $\mathbf{2.46e-003}$ | $1.72e+003$ | $1.73e+003$ | $1.72e+003$ | $1.72e+003$ | $4.13e+003$ | $1.92e+003$ |
| $f_9$ | Std | $\mathbf{5.51e-003}$ | $2.39e-011$ | 3.30 | $3.02e-001$ | $4.41e-008$ | $4.23e+002$ | $4.62e+001$ |
| | Rank | **1** | 3 | 5 | 2 | 3 | 7 | 6 |
| | Mean | $2.06e+001$ | $2.07e+001$ | $2.04e+001$ | $\mathbf{2.01e+001}$ | $2.07e+001$ | $2.05e+001$ | $2.08e+001$ |
| $f_{10}$ | Std | $5.22e-002$ | $1.01e-001$ | $6.10e-002$ | $\mathbf{3.82e-001}$ | $4.04e-002$ | $1.40e-001$ | $1.69e-002$ |
| | Rank | 3 | 5 | 2 | **1** | 5 | 4 | 7 |
| **Total Rank** | | **9** | **19** | **27** | **8** | **20** | **24** | **32** |

Table 5: Representation of an individual solution $i$. Nr is the maximum number of RFID readers that are deployed in the working area.

| | Reader 1 variables | | | Reader 2 variables | | | … | Reader $N_r$ variables | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Solution $i$ | $X_i^1$ | $Y_i^1$ | $P_i^1$ | $X_i^2$ | $Y_i^2$ | $P_i^2$ | … | $X_i^{N_r}$ | $Y_i^{N_r}$ | $P_i^{N_r}$ |

readers in the working area in order to achieve five goals described in Section 4.1. Figure 7 shows an example of a working area containing 100 RFID tags and 1 RFID reader, where the following three decision variables are chosen in this work:

> $X$: the $x$-axis coordinate value of the RFID reader;
>
> $Y$: the $y$-axis coordinate value of the RFID reader;
>
> $P$: the read range (i.e., radiate power level) of the RFID reader.

These variables can be encoded into solution's representation shown in Figure 7. We employ a representation in which each solution is characterized by a $D = 3N_r$ ($N_r$ is the total number of readers are that deployed in the network) dimensional real number vector. In the representation, $2N_r$ dimensions indicate the coordinates of the readers in the 2-dimensional working area, and the other $1N_r$ dimensions denote the interrogation range of each reader (which is determined by the radiated power).

*5.2. Implementation of the HABC Algorithm for the RNP Problem.* To apply the HABC algorithms to solve the RNP problem, the following steps should be taken and repeated (Figure 4).

*Step 1* (RFID deployment parameters initialization). The deployment parameters consist of reader control variables and RFID networks topology. The former include the adjustable radiated power range, the corresponding recognition scope—the distance up to which tag can be read by the reader, the interference range—the distance within which reader collision mainly occurs.

The networks topology includes the shape and dimension of the region, the number of the RFID tags to be used, the tag distribution (i.e., the tag position) in the working area, and the tag power threshold—the minimum tag received power level under which the communication between reader and tag can be established.

*Step 2* (encoding). Readers' variables consisting of the position and radiated power range should be encoded into the algorithm individual's representation as Table 5. The boundary limit of solution is defined by the networks topology.

*Step 3* (population generation). Produce the initial HABC population. Initialize $N$ species; each is divided into $K$ subpopulation. Each subpopulation $P_{ij}$ possesses $M$ individuals, where $i \in [1, \ldots, N]$, $j \in [1, \ldots, K]$. Then $N \times M \times K$ ($N \geq 2, M \geq 2, K \geq 1$) individuals based on $D$-dimensional objective should be randomly generated as shown in Figure 17, where $x_{ijks}$ ($i \in [1, \ldots, M]$, $j \in [1, \ldots, S]$, $k \in [1, \ldots, K]$, $s \in [1 \ldots N]$, $S = D/K$) is the position of the $j$th state variable in the $i$th individual of $k$th subpopulation of the $s$th species. $K$ is the group number by dividing $D$-dimensions into $S$-dimension. Emphasize that the group number $K$ is dynamically changed by random grouping approach by (11).

Notice that, each individual is characterized has a dimension $D$ equal to $3N$ ($N$ is the number of used RFID readers, $N = 10$ in this case), in which $2N$ dimensionalities for

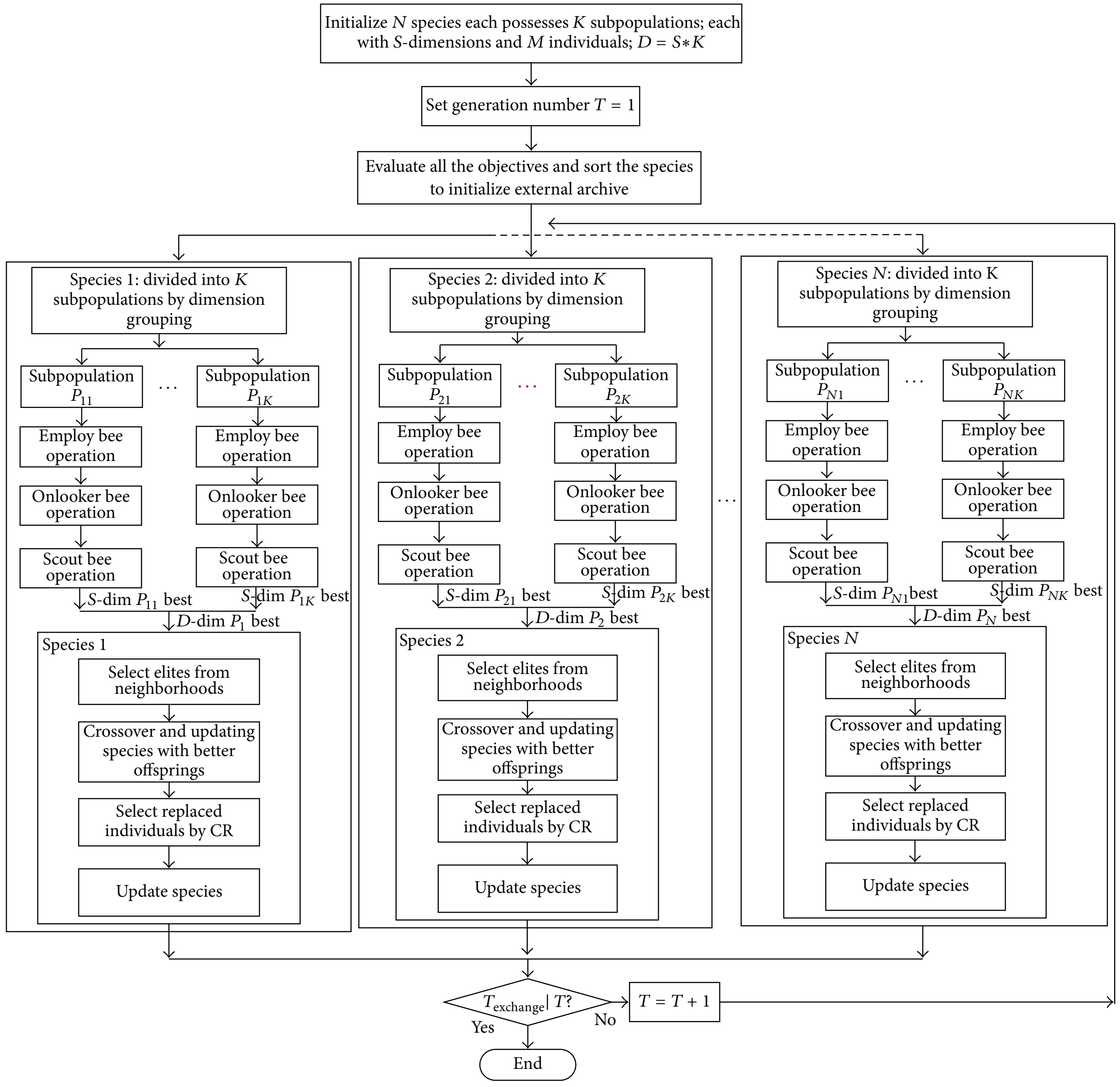

Initialize $N$ species each possesses $K$ subpopulations; each with $S$-dimensions and $M$ individuals; $D = S*K$
Set generation number $T = 1$
Evaluate all the objectives and sort the species to initialize external archive
Species 1: divided into $K$ subpopulations by dimension grouping
Subpopulation $P_{11}$
...
Subpopulation $P_{1K}$
Employ bee operation
Onlooker bee operation
Scout bee operation
$S$-dim $P_{11}$ best
$S$-dim $P_{1K}$ best
$D$-dim $P_1$ best
Species 1
Select elites from neighborhoods
Crossover and updating species with better offsprings
Select replaced individuals by CR
Update species
Species 2: divided into $K$ subpopulations by dimension grouping
Subpopulation $P_{21}$
Subpopulation $P_{2K}$
$S$-dim $P_{21}$ best
$S$-dim $P_{2K}$ best
$D$-dim $P_2$ best
Species 2
Species $N$: divided into K subpopulations by dimension grouping
Subpopulation $P_{N1}$
Subpopulation $P_{NK}$
$S$-dim $P_{N1}$best
$S$-dim $P_{NK}$ best
$D$-dim $P_N$ best
Species $N$
$T_{\text{exchange}} | T$?
Yes
No
$T = T + 1$
End

Figure 4: Flowchart of the HABC algorithm.

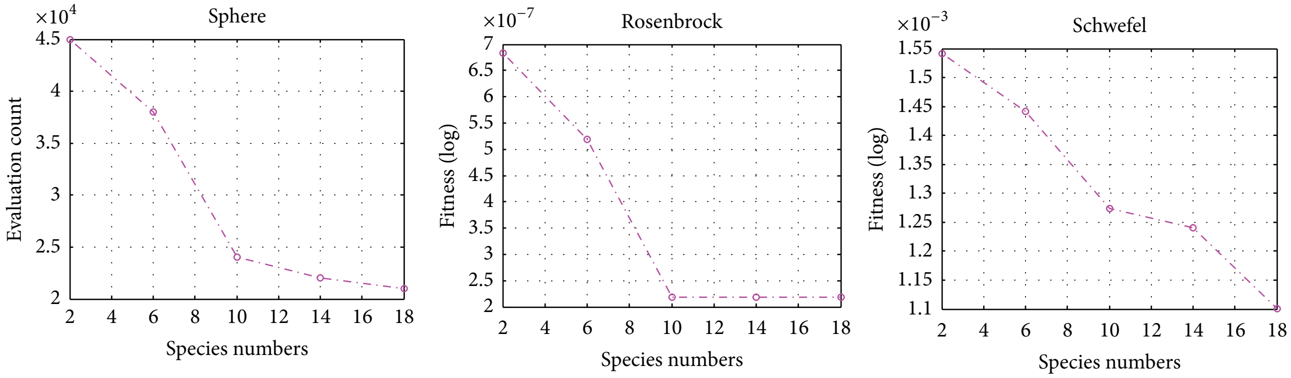
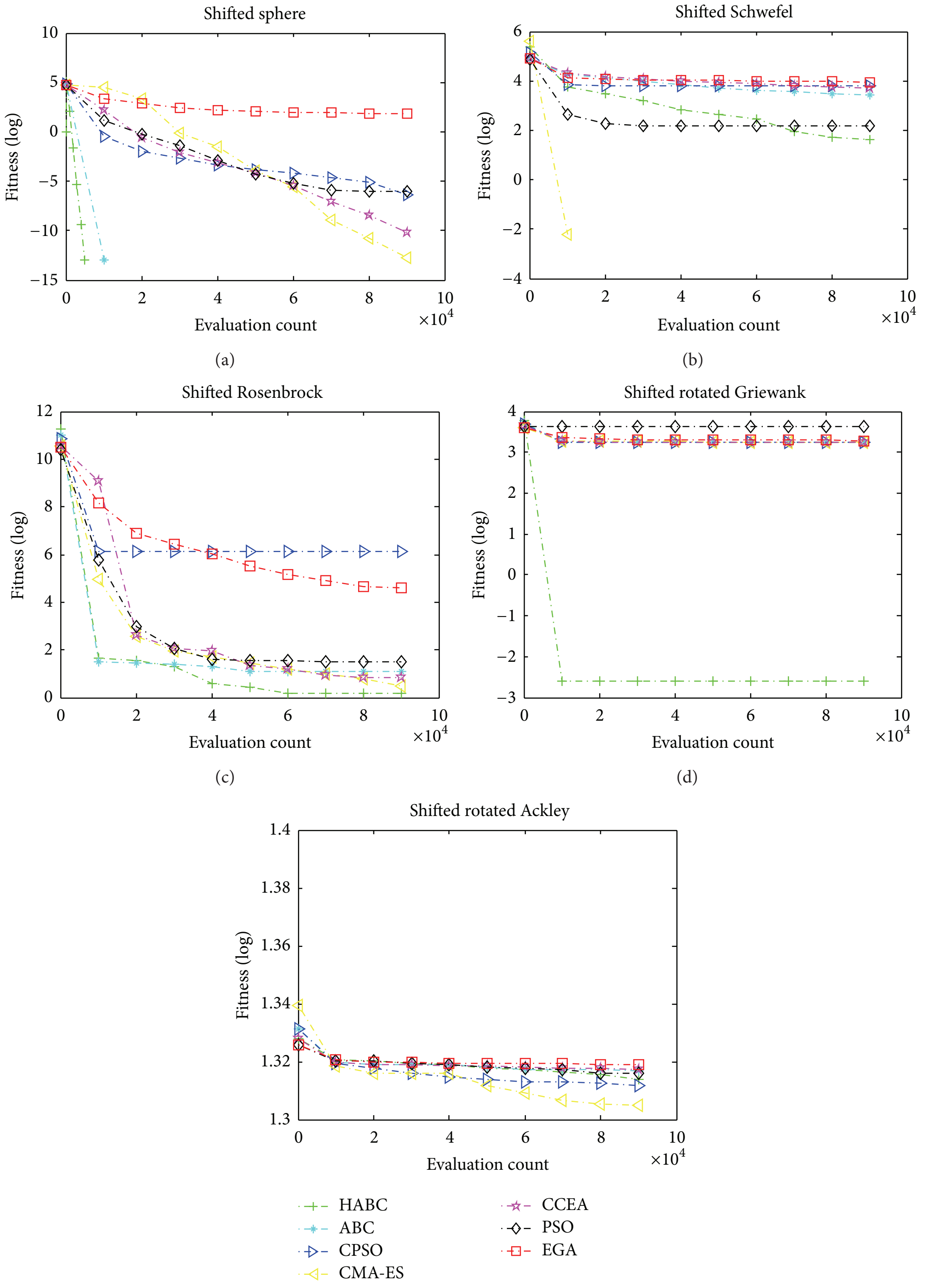


Figure 5: Results on test functions with different species number $N$.



(a)

(b)

(c)

(d)

(e)

FIGURE 6: The median convergence results on 100*D* CEC functions. (a) Shifted sphere function; (b) shifted Schwefel's function; (c) shifted Rosenbrock's function; (d) shifted rotated Griewank's function; (e) shifted rotated Ackley's function.

Table 6: The example for Cd100, Rd500 and Rd500.

| | Reader Specification | | Topology Specification | |
|---|---|---|---|---|
| Cd100 | Reader number | 10 | Dimension | 30 m × 30 m |
| | Radiated power | 0.1–2 watt | Tag number | 100 |
| | Interrogation range | 3-4 m | Tag distribution | Uniform |
| | Interference range | 3.5–4.5 m | Tag power threshold | −10 dBm |
| Rd500 | Reader number | 50 | Dimension | 150 m × 150 m |
| | Radiated power | 0.1–2 watt | Tag number | 500 |
| | Interrogation range | 3-4 m | Tag distribution | Uniform |
| | Interference range | 3.5–4.5 m | Tag power threshold | −10 dBm |

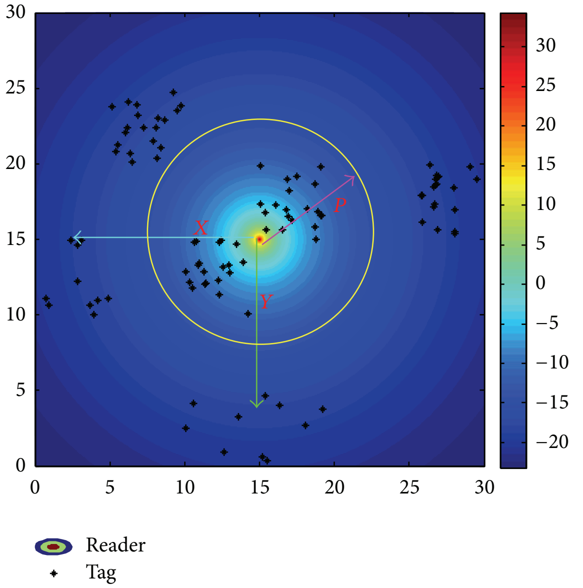


Figure 7: Example of a 30 m × 30 m working area with one RFID reader and 100 tags. The radiated power level (dBm) marked on the right diagram sidebar and the yellow cycle represents the read range of the RFID reader.

the coordinates of reader positions, and 1$N$ dimensionalities for radiated powers of each reader.

*Step 4* (construct complete evolving solution $G_{\text{best}}$). $G_{\text{best}}$ is the concatenation of the best subcomponents' solutions $P_{ij}$.

Set cycle = 1.

*Step 5* (optimization procedure). Loop over each species $P_i$, $i \in [1, \ldots, N]$: for each subpopulation $P_{ij}$, $j \in [1, \ldots, K]$, do the following:

(1) fitness calculation:

for each individual $X_i$, $i \in [1, \ldots, M]$, replace the $i$th component of the $G_{\text{best}}$ by using the $i$th component of $X_i$.

Calculate the new solution fitness: $f(\text{new}\,G_{\text{best}}(P_1 \cdot g, P_2 \cdot g, X_i, \ldots, P_k \cdot g))$ using (1)–(5).

If $f(\text{new}\,G_{\text{best}}) < f(G_{\text{best}})$, then $G_{\text{best}}$ is replaced by new $G_{\text{best}}$;

(2) update $X_{iks}$ positions using (8);

(3) memorize the best solution achieved so far, Compare the best solution with $G_{\text{best}}$ and memorize the better one;

(4) select elites to the best-performing list (BPL) from current species' neighborhood.

(5) crossover and mutation between species by (12);

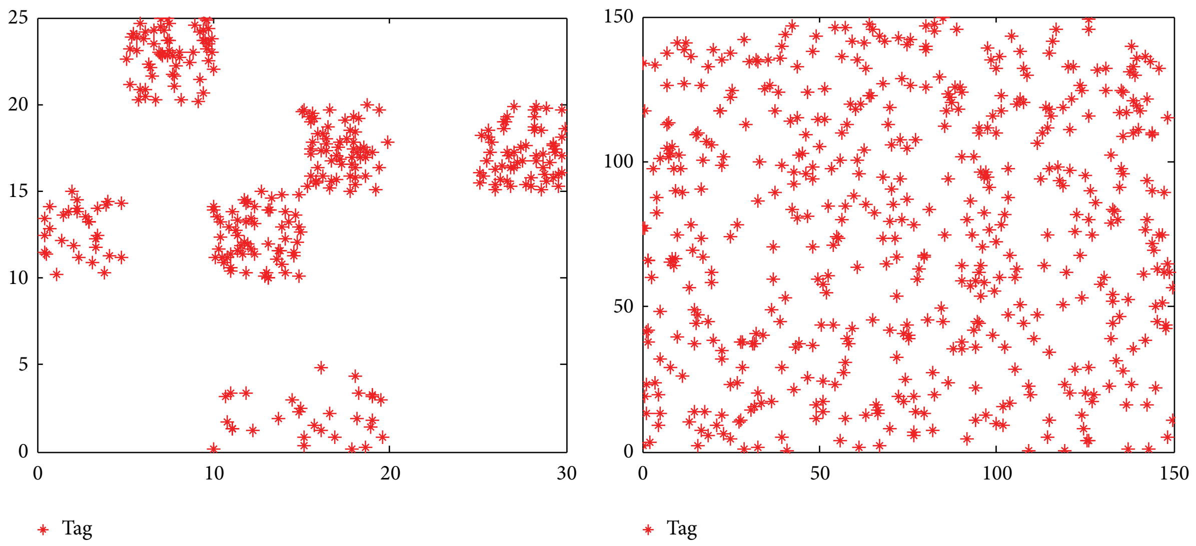
(6) update species with greedy selection strategy;

cycle = cycle +1.

*Step 6* (termination condition). If the cycle is greater than limited value, stop the procedure; otherwise, go to Step 5.

*5.3. Simulation Configurations.* The readers used here are mobile and the tags are passive. According to the references [6, 33–37], the related RFID readers' parameters can be set as in Table 6. Here the interrogation range according to the reader radiated power is computed as in [38]. The proposed algorithm is evaluated against two different RNP instances, namely, Cd100 and Rd500. The instance of Cd100 is tested on a 30 m × 30 m working space with 100 clustered distributed tags. Another instance, namely, Rd500, contains 500 randomly distributed tags in a 150 m × 150 m working space (shown in Figure 8). In this experiment, the parameters setting for HABC, ABC, PSO, EGA, and CCEA can be the same as in Section 4.1. Especially, the PS$^2$O algorithm proposed by us in [6] as an effective approach for solving RNP, is employed to compare with the proposed approach using HABC. For PS$^2$O, the number of swarms and swarm size can be set by $n = 10$, $m = 5$. The constriction factor is used with $\chi = 0.729$, and then the learning rates $c_1 = c_2 = c_3 = 1.3667$ [6].

*5.4. Results of the RNP with Cd100.* In this section, an RNP instance, called Cd100, in which 10 readers are deployed in the 30 m×30 m working space with 100 clustered distributed tags, is employed, which can be considered as a continuous optimization problem with 30 dimensions, shown in Figure 8(a). All algorithms are firstly tested on the four objective functions ($f_c$, $f_i$, $f_e$, and $f_l$), respectively. In the single objective



(a)

(b)

Figure 8: The test working area tags deployment: (a) cluster deploying with 100 tags (Cd100); (c) randomly deploying with 500 tags (Rd500).

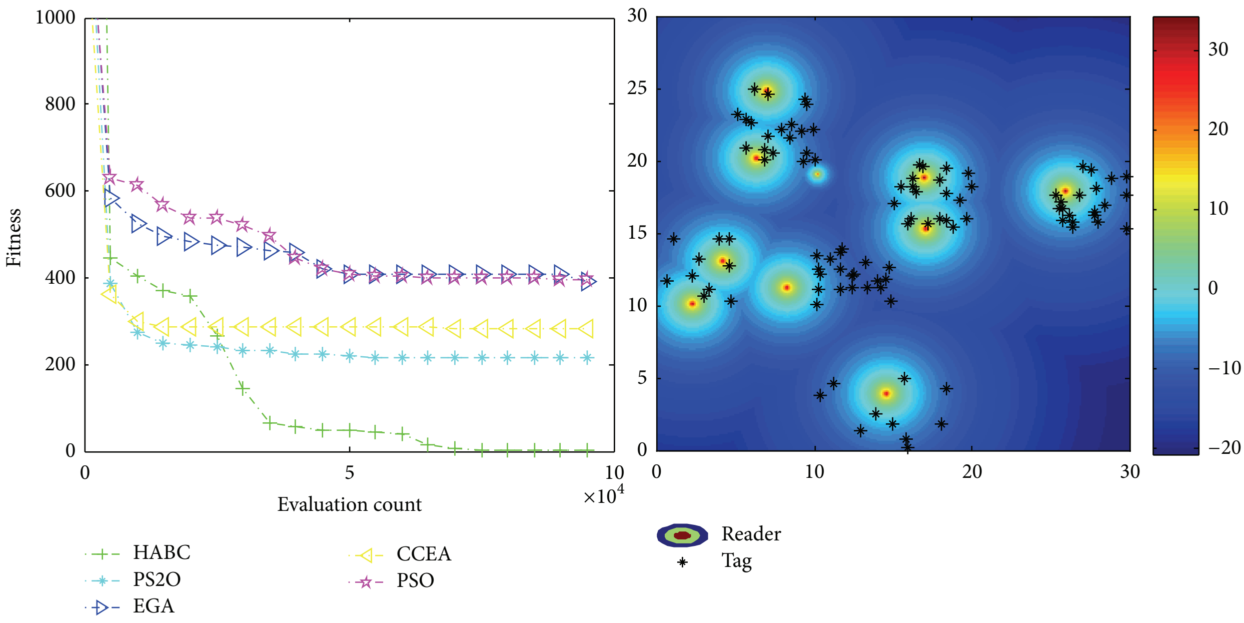
Table 7: Performance of all algorithms on RNP with Cd100.

| Objective Func. | | HABC | PS$^2$O | EGA | CCEA | PSO |
|---|---|---|---|---|---|---|
| $f_c$ | Best | 218.2832 | **216.4532** | 359.0912 | 233.3948 | 390.0938 |
| | Worst | **247.1265** | 268.2351 | 411.9823 | 334.3984 | 400.8675 |
| | Mean | **221.0934** | 241.4512 | 389.4652 | 282.9834 | 395.8343 |
| | Std | **80.3842** | 102.4323 | 91.0912 | 83.2387 | 130.3323 |
| $f_i$ | Best | 144.0932 | **142.0934** | 147.3454 | 145.8723 | 150.0923 |
| | Worst | 146.8812 | **145.6421** | 152.4325 | 156.0912 | 163.6433 |
| | Mean | 145.9823 | **144.4326** | 149.6783 | 148.8542 | 155.0933 |
| | Std | **3.8721** | 5.9564 | 9.8123 | 14.9812 | 26.0232 |
| $f_e$ | Best | **2.1583e − 015** | 2.5612e − 014 | 2.6234e − 01 | 2.1678e − 014 | 2.6563 |
| | Worst | **2.8923e − 015** | 3.3412e − 014 | 3.6973e − 01 | 4.09823e − 014 | 3.4938 |
| | Mean | **2.5812e − 015** | 5.5434e − 014 | 3.1034e − 01 | 2.7023e − 014 | 3.1276 |
| | Std | **2.58923e − 011** | 5.1654e − 014 | 3.8712e − 01 | 4.9898e − 011 | 2.6512 |
| $f_l$ | Best | **5.2401e − 0020** | 3.0934e − 013 | 2.9845e − 01 | 6.3412e − 009 | 3.3675 |
| | Worst | **3.2467e − 0019** | 16.883 | 0.0045 | 4.3658e − 008 | 17.0912 |
| | Mean | **1.3935e − 0019** | 3.9812 | 0.0009 | 2.4519e − 008 | 9.9814 |
| | Std | **6.6902e − 0020** | 5.7652 | 0.0022 | 7.0914e − 009 | 4.6575 |
| $f_m$ | Best | **0.1798** | 0.2351 | 0.1864 | 0.1994 | 0.2123 |
| | Worst | **0.2254** | 0.3566 | 0.2429 | 0.2411 | 0.2758 |
| | Mean | **0.1980** | 0.2871 | 0.2105 | 0.2118 | 0.2394 |
| | Std | **0.1314** | 0.3720 | 0.1556 | 0.1441 | 0.2173 |

optimization, the results are providing an optimal solution for a single objective that does not take account of the others. After that, the test of the combined objective function $f_m$ is implemented. The weighted coefficients used in this instance are set as $w_1 = 0.35$, $w_2 = 0.15$, $w_3 = 0.2$, and $w_4 = 0.3$, which also can be varied according to the demand of different network. The results consisting of the best, worst, mean, and standard deviation of the optimal solutions over 50 sample runs are listed in Table 7.

From Table 7, it is observed that the HABC can get better results in most proposed objective functions in comparison to other algorithms (PS$^2$O, EGA, CCEA, and PSO), except $f_i$. Particularly, the good performance in the combined objective function $f_m$, suggests that the proposed approach using HABC outperforms the other algorithms in optimizing the models presented in this paper.

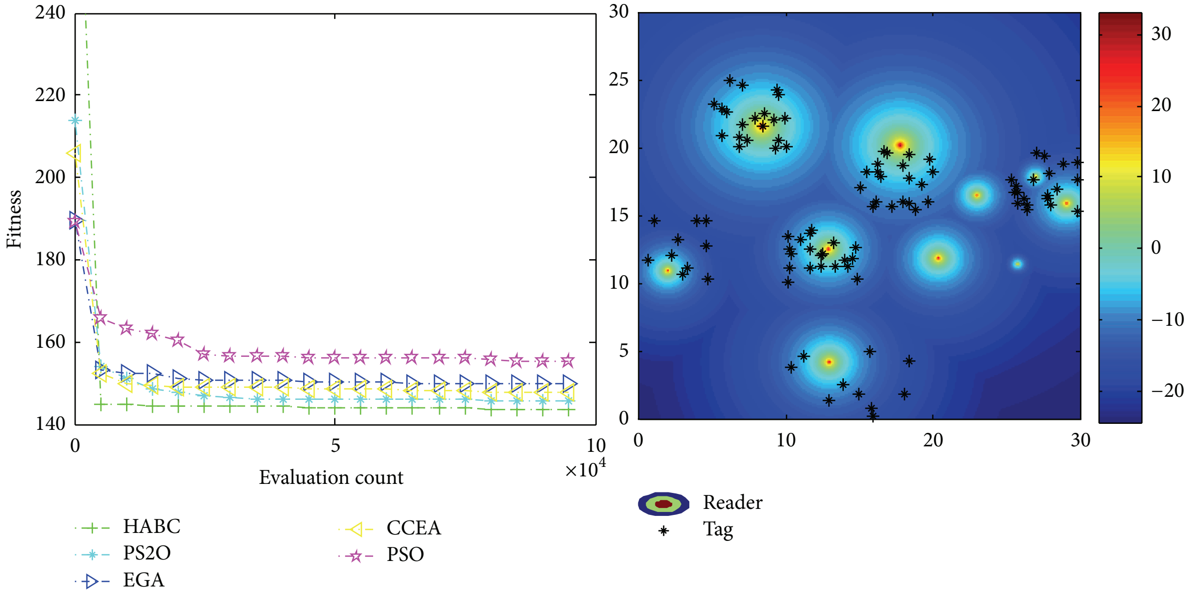
Figure 9 illustrates the result only considering the coverage of readers. Figure 9(a) gives the convergence process of



(a)

(b)

Figure 9: Results only consider tag coverage. (a) Convergence process; (b) reader location and received power distribution.



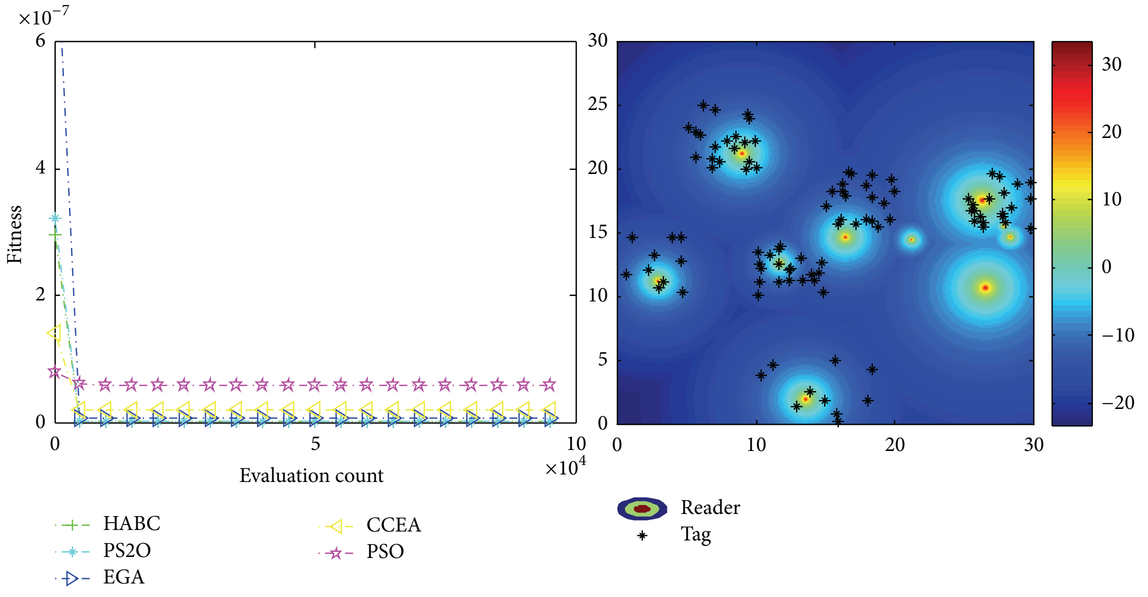
(a)

(b)

Figure 10: Results only consider interference. (a) Convergence process; (b) reader location and received power distribution.

the average values obtained by HABC and other algorithms over 50 sample runs for the objective function $f_c$. The corresponding reader locations and the distribution of their radiated power optimized by HABC are shown in Figure 9(b). In this case, according to the demand of higher tag coverage, the algorithms adjust the power and balance the deployment of readers in the working area. From Figure 9(a), the HABC has a faster convergence and gets better results than the other three algorithms. From Figure 9(b), it is obviously observed that the HABC can schedule with the reader network with higher tag coverage.

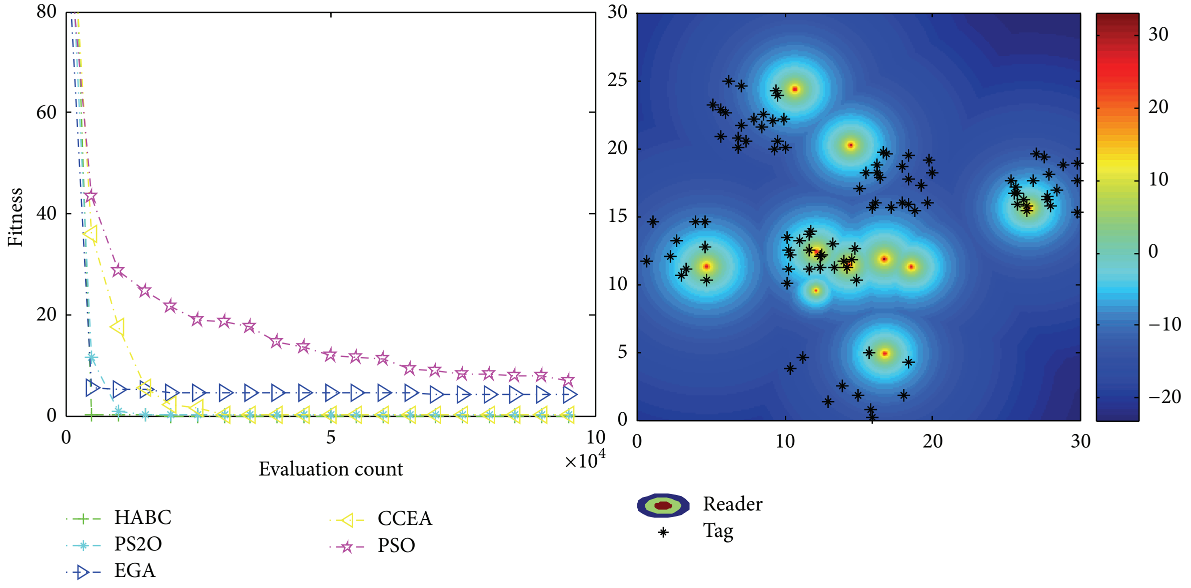
Note that the visual results of the RNP with other four single objects have similar trends, as shown in Figures 10–14. Figure 10 illustrates the result implemented by all algorithms with only considering the interference between readers. In this optimization mode, the algorithms aim to maintain sufficient distances between RFID readers. We can observe from Figure 10(b) that network operating condition



(a)

(b)

Figure 11: Results only consider economic efficiency. (a) Convergence process; (b) reader location and received power distribution.



(a)

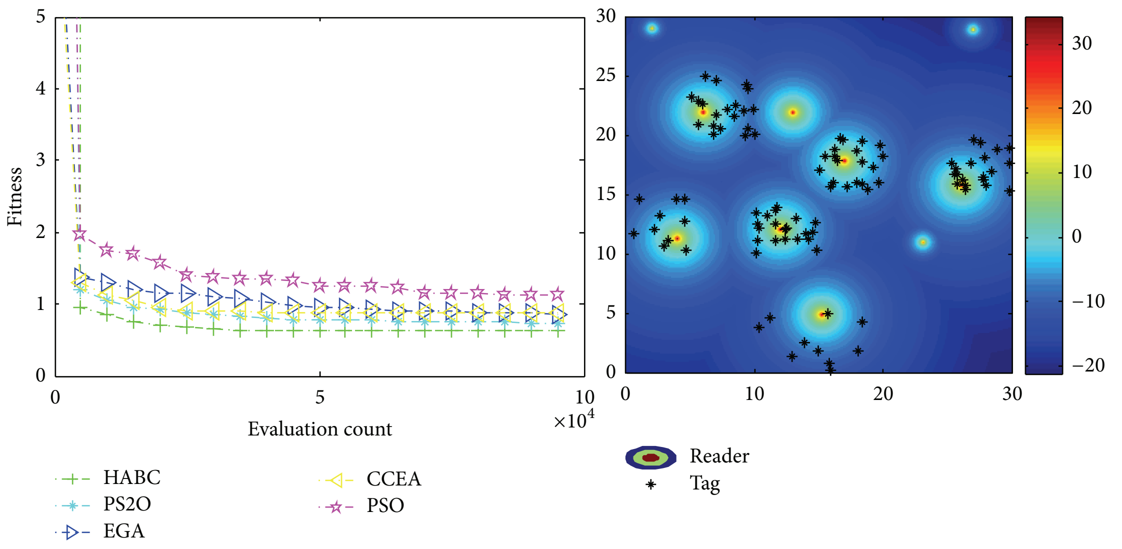
(b)

Figure 12: Results only consider load balance. (a) Convergence process; (b) reader location and received power distribution.

deteriorates, without guaranteeing complete tag coverage rate while the objective function $f_i$ is perfectly optimized, because the readers are optimized to move away from each other and thus located far from high traffic areas for the purpose of minimizing the reader interference.

Similarly, Figures 11 and 12 show the results considering economic efficiency and load balance, respectively. In the optimization mode with only considering economic efficiency, the algorithms aim to locate the readers close to the cluster center of the tag dense areas. From Figures 11(a) and 11(b), it is clear to see that the HABC algorithm gets significant superiority in the convergence and accuracy of solutions. From Figure 12, HABC outperforms all the other algorithms in the optimization mode with balancing the number of tags served by each reader and radiated power of the reader.

Figure 13(a) shows the convergence process of all algorithms with the combined objective function $f_m$ when all the requirements are considered. We can observe from



(a)

(b)

Figure 13: Results of the combined measurement. (a) Convergence process; (b) reader location and received power distribution.

Table 8: Performance of all algorithms on RNP with Cd500.

| Objective Func. | | HABC | $PS^2O$ | EGA | CCEA | PSO |
|---|---|---|---|---|---|---|
| $f_c$ | Best | **723.3412** | 761.4376 | 931.5424 | 965.1634 | 877.0912 |
| | Worst | **951.312** | 1.024e + 003 | 1.9652e + 003 | 998.8912 | 1.1274e + 003 |
| | Mean | **832.4521** | 981.9812 | 1.21654e + 003 | 981.9812 | 983.3461 |
| | Std | 65.3476 | 110.2360 | 361.0912 | **61.1653** | 141.7735 |
| $f_i$ | Best | **530.6258** | 545.5313 | 539.9123 | 531.6734 | 584.4458 |
| | Worst | **541.9812** | 573.4476 | 568.3476 | 562.6423 | 662.9891 |
| | Mean | 546.3155 | 558.0246 | 541.6712 | **545.7813** | 600.2657 |
| | Std | **4.8912** | 9.2376 | 14.3452 | 8.6791 | 15.9824 |
| $f_e$ | Best | **1.0126e − 08** | 3.2345e − 008 | 4.2351e − 007 | 8.7342e − 007 | 3.6342e − 007 |
| | Worst | **1.7981e − 08** | 5.1852e − 008 | 9.8782e − 007 | 9.7674e − 007 | 4.4487e − 007 |
| | Mean | **1.4621e − 08** | 4.3186e − 008 | 7.4127e − 007 | 9.2412e − 007 | 5.1956e − 007 |
| | Std | **1.5431e − 09** | 6.3671e − 008 | 8.9834e − 007 | 9.8671e − 008 | 3.6423e − 008 |
| $f_l$ | Best | **6.2234e − 0018** | 7.4523412e − 007 | 4.0934e − 08 | 4.9834e − 07 | 6.4867 |
| | Worst | **4.2823e − 0017** | 5.4123e − 006 | 25.5623 | 0.0085 | 24.1438 |
| | Mean | **2.3923e − 0017** | 3.5978e − 006 | 8.0921 | 0.0069 | 10.5643 |
| | Std | **6.6574e − 0018** | 7.4123e − 007 | 8.7321 | 0.0067 | 5.8904 |
| $f_m$ | Best | **0.2549** | 0.3640 | 0.3365 | 0.3521 | 0.5051 |
| | Worst | **0.2637** | 0.3935 | 0.4454 | 0.4633 | 0.5322 |
| | Mean | **0.2563** | 0.3780 | 0.4067 | 0.4232 | 0.5168 |
| | Std | **0.0785** | 0.0412 | 0.3223 | 0.3521 | 0.0106 |

Figure 13(b) that HABC still finds the best solution, which is a reasonable compromise between different demands. Moreover, the convergence process of the five objective functions has a similar varying tendency, proving that the proposed HABC has faster convergence rate than other algorithms.

*5.5. Results of the RNP with Rd500.* Apparently, with the increasing number of the deployed readers and tags in the working area, the complexity of solving the RNP problem increases exponentially. Therefore, to further verify the efficiency of the proposed algorithm, the instance with larger scales, namely, Rd500, is also employed: 50 readers are deployed in the 150 m × 150 m working space with 500 tags that are randomly distributed, which can be considered as a continuous optimization problem with 150 dimensions, shown in Figure 8(b). Similar to the test in Section 5.4, the proposed approach using HABC and other algorithms is firstly tested with the four single objects presented in (1)–(5),

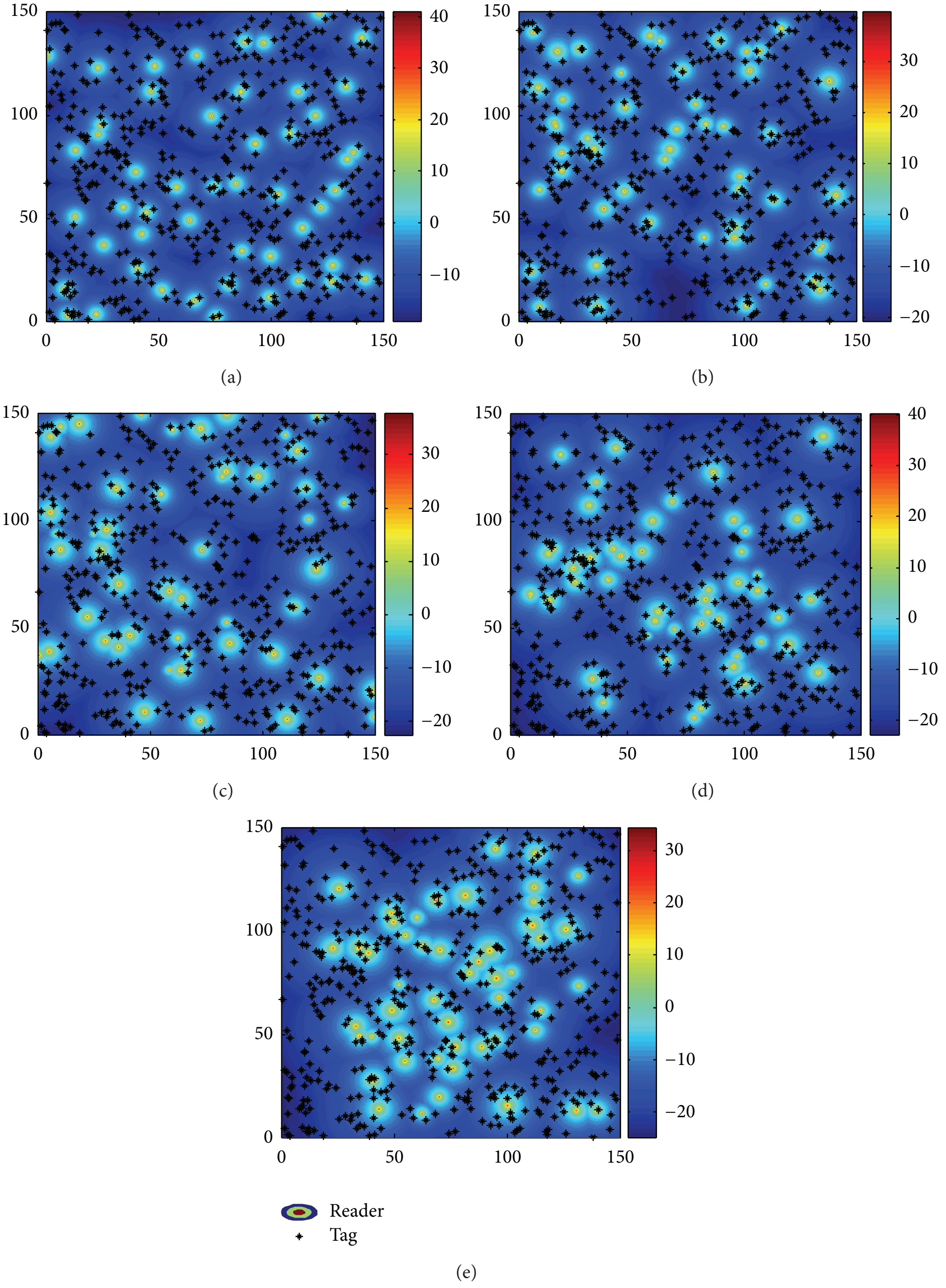


Figure 14: The results of all algorithms on Cd500 with the combined objective function $f_m$. (a) Result of HABC; (b) result of PS$^2$O; (c) result of EGA; (d) result of CCEA; (e) result of PSO.

respectively. Then the test of the combined objective function is implemented. The weighted coefficients are the same as in Section 5.4. All the testing results are listed in Table 8 over 50 sample runs. As shown in Table 8, the results demonstrate that the proposed approach using HABC obtains superior solutions consisting of best, worst, and mean values to those of other algorithms on all the five objective functions, better than the performance of HABC tested on Cd100 instance in which HABC outperforms other algorithms on three of five objective functions. As expected, by employing the decomposing strategy of HABC, the networks problem can be divided into several smaller ones to reduce the computational complexity, achieving better results.

The reader locations and the distribution of their radiated power optimized by all algorithms are shown in Figures 14(a)–14(e), respectively. The results in Figure 14 demonstrate

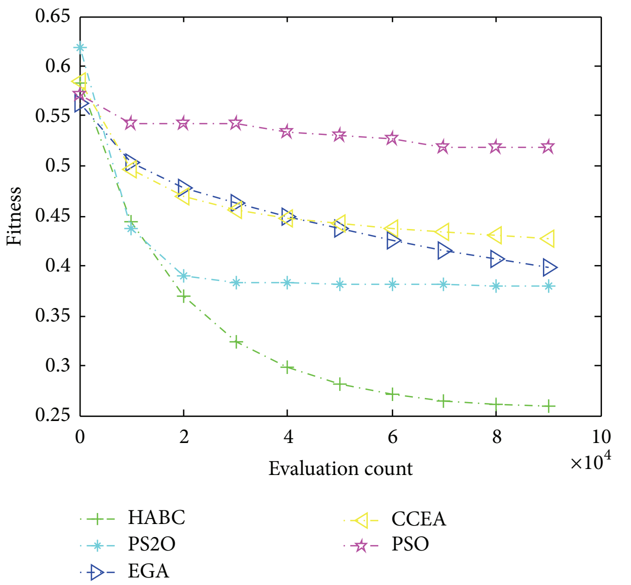


FIGURE 15: The convergence process of all algorithms with the combined objective function $f_m$ on Cd500.

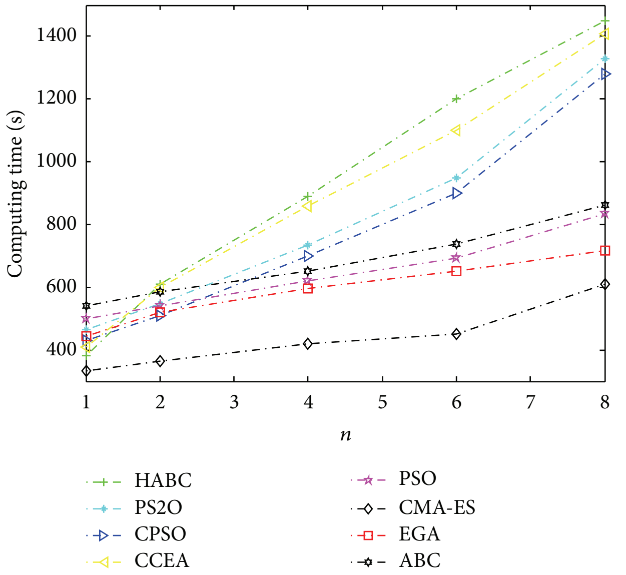


FIGURE 16: Computing time of all algorithms on different RNP problems.

that the proposed approach using HABC can ensure a reasonable deployment in the aspects of maximizing the tag coverage, minimizing the interference between readers, minimizing the distances between readers and tags cluster centers, optimizing the load balance, and outperforming the compared algorithms. Figure 15 shows that the HABC still has a faster convergence progress than other algorithms in handling larger-scale problem.

*5.6. Algorithm Complexity Analysis.* Algorithm complexity analysis is presented briefly as follows. If we assume that the computation cost of one individual in the HABC is Cost_$a$, the cost of the crossover operator is Cost_$c$ and the total computation cost of HABC for one generation is $N * K * M *$ Cost_$a$ + $N *$ Cost_$c$. However, because the heuristic algorithms used in this paper cannot ensure comprehensive convergence, it is very difficult to give a brief analysis in terms of time for all algorithms. By directly evaluating the algorithmic time response on different objective functions, the average computing time in 30 sample runs of all algorithms is given in Figure 16. From the results in Figure 16, it is observed that the HABC takes the most computing time in all compared algorithms and the time increasing rate of it is the highest one. This can explain that the multi-population cooperative co-evolution strategy integrated by HABC enhanced the local search ability at cost of increasing the computation amount. In summary, it is concluded that compared with other algorithms, the CMOABC requires more computing time to achieve better results.

## 6. Conclusion

In this paper, we present an optimization model for planning the positions and radiated power setting of readers in the RFID network. The four nonlinear RNP objective functions are formulated by considering tag coverage, reader interference, economic efficiency, and network load balance as the primary requirements of the real-world RFID system. And the combined measure is also given so that the multiple objectives can be optimized simultaneously.

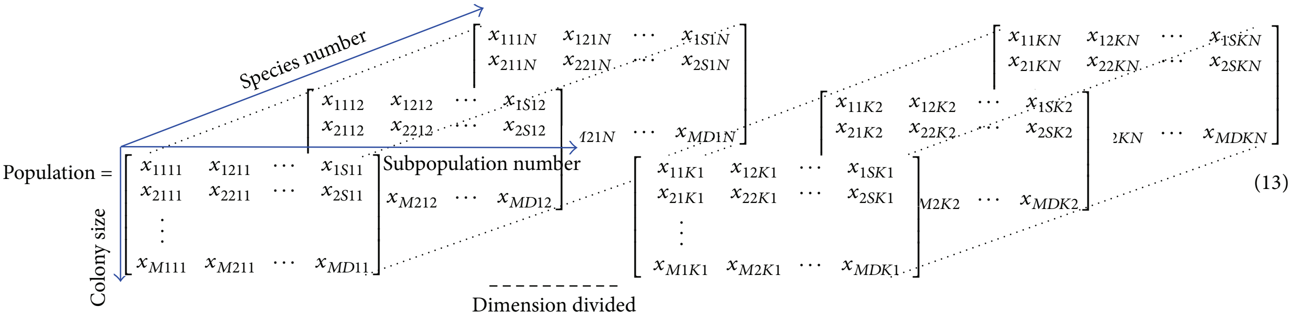
Finally, in order to solve the RNP model, a novel hierarchical artificial colony algorithm, called HABC, is proposed by extending single artificial bee colony (ABC) algorithm to hierarchical and cooperative mode by combining the multi-population cooperative co-evolution approach based on vector decomposing strategy and the comprehensive learning method. Results obtained from the proposed approach have been compared with those obtained by ABC, PSO, CPSO, EGA, CMA_ES, and CCEA. The experiment results show that for all the test functions the HABC gets significant superiority to the other six algorithms. HABC is then employed to solve the real-world RNP problem on two different-scale instances, namely, Cd100 and Rd500. The simulation studies show that the HABC remarkably outperforms other algorithms. Especially, in tackling larger-scale RNP problem (i.e., Rd500 instance), the HABC performs more effectively, indicating that the HABC is more suitable for solving high-dimension RNP problem.

## Highlights

(i) Hierarchical structure and cooperative coevolution ensure the local search ability of the HABC.

(ii) The concept of crossover and mutation operator is used in HABC.

(iii) The HABC is proved to have better performance than the PSO, ABC, CPSO, EGA, CCEA, and CMA_ES on benchmarks.



(13)

Figure 17

Table 9: Parameters of the test functions.

| $f$ | Dimensions | Initial Range | $x^*$ | $f(x^*)$ |
|---|---|---|---|---|
| $f_1$ | 100 | $[-100, 100]^D$ | [0, 0, . . ., 0] | 0 |
| $f_2$ | 100 | $[-30, 30]^D$ | [1, 1, . . .,1] | 0 |
| $f_3$ | 100 | $[-5.12, 5.12]^D$ | [0, 0, . . ., 0] | 0 |
| $f_4$ | 100 | $[-500, 500]^D$ | [420.9867, . . ., 420.9867] | 0 |
| $f_5$ | 100 | $[-32.768, 32.768]^D$ | [0, 0, . . ., 0] | 0 |
| $f_6$ | 100 | $[-100, 100]^D$ | [0, 0, . . ., 0] | −450 |
| $f_7$ | 100 | $[-100, 100]^D$ | [0, 0, . . ., 0] | −450 |
| $f_8$ | 100 | $[-100, 100]^D$ | [0, 0, . . ., 0] | 390 |
| $f_9$ | 100 | No bounds | [0, 0, . . ., 0] | −180 |
| $f_{10}$ | 100 | $[-32, 32]^D$ | [0, 0, . . ., 0] | −140 |

(iv) An RNP optimization model is proposed based on novel HABC algorithm.

(v) HABC obtains better optimal solutions than PS$^2$O, PSO, CCEA, and EGA on Cd100 and Rd500 instances.

(vi) HABC has considerable potential for solving high-dimensional optimization problems.

## Appendix

## List of Test Functions

These benchmark functions consist of the basic benchmark continuous functions $f_1 \sim f_2$ (unimodal), $f_3 \sim f_5$ (multimodal), and the shifted and rotated functions $f_6$–$f_{10}$ (CEC2005).

*Unimodal Benchmark Function*

(1) Sphere's function:

$$f_1(x) = \sum_{i=1}^{n} x_i^2. \tag{A.1}$$

(2) Rosenbrock's function:

$$f_2(x) = \sum_{i=1}^{n} 100 \times \left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2. \tag{A.2}$$

*Multimodal Benchmark Function*

(3) Rastrigrin function

$$f_5(x) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos\left(2\pi x_i\right)\right) + 10. \tag{A.3}$$

(4) Schwefel's function:

$$f_6(x) = D * 418.9829 + \sum_{i=1}^{D} - x_i \sin\left(\sqrt{\left|x_i\right|}\right). \tag{A.4}$$

(5) Ackley's function:

$$\begin{aligned} f_7(x) = & -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) \\ & - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i\right) + 20 + e. \end{aligned} \tag{A.5}$$

*CEC2005 Function*

(6) Shifted Sphere's function:

$$f_9(x) = \sum_{i=1}^{D} z_i^2 + f_{\text{bias1}}, \quad z = x - o. \tag{A.6}$$

(7) Shifted Schwefel's Problem 1.2:

$$f_{10}(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2 + f_{\text{bias}}, \quad z = x - o. \tag{A.7}$$

(8) Shifted Rosenbrock's function:

$$f_{11}(x) = \sum_{i=1}^{D-1}\left(100\left(z_i^2 - z_{i+1}\right)^2 + \left(z_i^2 - 1\right)^2\right) + f_{\text{bias}}. \tag{A.8}$$

(9) Shifted rotated Griewank's function without bounds:

$$f_{12}(x) = \sum_{i=1}^{D}\frac{{z_i}^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}}. \tag{A.9}$$

(10) Shifted rotated Ackley's function with global optimum on bounds:

$$\begin{aligned} f_{13}(x) = & -\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}{z_i}^2}\right) \\ & -\exp\left(\sum_{i=1}^{D}\cos\left(2\pi z_i\right)\right) + 20 \\ & + e + f_{\text{bias}}, \quad z = (x - o) * M. \end{aligned} \tag{A.10}$$

*Parameters of the Test Functions*. The dimensions, initialization ranges, global optima $x^*$, and the corresponding fitness value $f(x^*)$ of each function are listed in Table 9.

## Conflict of Interests

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

## Acknowledgment

This work is supported in part by the Natural Science Foundation of China under Grant nos. 61105067 and 61174164.

## References

[1] T. H. Chang, S. C. Hsu, and T. C. Wang, "A proposed model for measuring the aggregative risk degree of implementing an RFID digital campus system with the consistent fuzzy preference relations," *Applied Mathematical Modelling*, vol. 37, no. 5, pp. 2605–2622, 2013.

[2] B. Carbunar, M. K. Ramanathan, M. Koyutürk, S. Jagannathan, and A. Grama, "Efficient tag detection in RFID systems," *Journal of Parallel and Distributed Computing*, vol. 69, no. 2, pp. 180–196, 2009.

[3] B. Carbunar, M. K. Ramanathan, M. Koyuturk, C. Hoffmann, and A. Grama, "Redundant reader elimination in RFID systems," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 176–184, Santa Clara, Calif, USA, September 2005.

[4] Q. Guan, Y. Liu, Y. P. Yang, and W. S. Yu, "Genetic approach for network planning in the RFID systems," in *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA '06)*, vol. 2, pp. 567–572, Jinan, China, October 2006.

[5] M. F. Tasgetiren, Q. K. Pan, P. N. Suganthan, and A. Oner, "A discrete artificial bee colony algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion," *Applied Mathematical Modelling*, vol. 37, no. 10-11, pp. 6758–6779, 2013.

[6] H. N. Chen, Y. L. Zhu, K. Y. Hu, and T. Ku, "RFID network planning using a multi-swarm optimizer," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 888–901, 2011.

[7] H. Rashidi and E. P. K. Tsang, "Novel constraints satisfaction models for optimization problems in container terminals," *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 3601–3634, 2013.

[8] Y. Gao, X. Hu, H. L. Liu, and Y. Feng, "Multiobjective estimation of distribution algorithm combined with PSO for RFID network optimization," in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '10)*, vol. 2, pp. 736–739, Changsha City, China, March 2010.

[9] Y. H. Yang, Y. J. Wu, M. Xia, and Z. J. Qin, "A RFID network planning method based on genetic algorithm," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, vol. 1, pp. 534–537, Wuhan, China, April 2009.

[10] H. N. Chen and Y. L. Zhu, "RFID networks planning using evolutionary algorithms and swarm intelligence," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–4, Dalian, China, October 2008.

[11] J. H. Seok, J. Y. Lee, C. Oh, J. J. Lee, and H. J. Lee, "RFID sensor deployment using differential evolution for indoor mobile robot localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 3719–3724, Taipei, Taiwan, October 2010.

[12] Y. J. Gong, M. Shen, J. Zhang, O. Kaynak, W. N. Chen, and Z. H. Zhan, "Optimizing RFID network planning by using a particle swarm optimization algorithm with redundant readerelimination," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 900–912, 2012.

[13] I. Bhattacharya and U. K. Roy, "Optimal placement of readers in an RFID network using particle swarm optimization," *International Journal of Computer Networks & Communications*, vol. 2, no. 6, pp. 225–234, 2010.

[14] L. B. Ma, K. Y. Hu, Y. L. Zhu, and H. N. Chen, "Computational evolution of social behavior in bacterial colony optimization model," *Journal of Pure & Applied Microbiology*, vol. 7, pp. 48–493, 2013.

[15] H. N. Chen, Y. L. Zhu, and K. Y. Hu, "Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 539–547, 2010.

[16] J. J. Yang and Z. M. Fei, "Bipartite graph based dynamic spectrum allocation for wireless mesh networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 96–101, Beijing, China, June 2008.

[17] J. J. Yang and Z. M. Fei, "A new channel assignment algorithm for wireless mesh networks," *International Journal of Pervasive*

*Computing and Communications*, vol. 5, no. 3, pp. 233–248, 2009.

[18] M. A. Potter and K. A. de Jong, "Cooperative coevolution: an architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.

[19] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to participle swam optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[20] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.

[21] M. N. Omidvar, X. Li, X. Yao, and Z. Yang, "Cooperative coevolution for large scale optimization through more frequent random grouping," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–8, Barcelona, Spain, July 2010.

[22] W. P. Zou, H. N. Chen, Y. L. Zhu, and B. W. Zhang, "Solving multiobjective optimization problems using artificial bee colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2011, Article ID 569784, 37 pages, 2011.

[23] D. Sofge, K. D. Jong, and A. Schultz, "A blended population approach to cooperative coevolution for decomposition of complex problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 413–418, Honolulu, Hawaii, USA, May 2002.

[24] D. W. Engels and S. E. Sarma, "The reader collision problem," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 370–376, Yasmine Hammamet, Tunisia, October 2002.

[25] Q. Dong, A. Shukla, V. Shrivastava, D. Agrawal, S. Banerjee, and K. Kar, "Load balancing in large-scale RFID systems," *Computer Networks*, vol. 52, no. 9, pp. 1782–1796, 2008.

[26] 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.

[27] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[28] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Foundations of Fuzzy Logic and Soft Computing*, vol. 4529 of *Lecture Notes in Computer Science*, pp. 789–798, Springer, Berlin, Germany, 2007.

[29] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[30] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[31] S. Sumathi, T. Hamsapriya, and P. Surekha, *Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*, Springer, Berlin, Germany, 1st edition, 2008.

[32] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[33] Q. Guan, Y. Liu, Y. P. Yang, and W. Yu, "Genetic approach for network planning in the RFID systems," in *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA '06)*, vol. 2, pp. 567–572, Jinan, China, October 2006.

[34] H. Feng and J. Qi, "Optimal RFID networks planning using a hybrid evolutionary algorithm and swarm intelligence with multi-community population structure," in *Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT '12)*, pp. 1063–1068, Pyeongchang, Republic of Korea, February 2012.

[35] H.-P. Huang and Y.-T. Chang, "Optimal layout and deployment for RFID systems," *Advanced Engineering Informatics*, vol. 25, no. 1, pp. 4–10, 2011.

[36] Y.-C. Wang and Y.-C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[37] X. W. Wang, J. R. Wu, and L. Guo, "A k-coverage algorithm in three dimensional wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT '10)*, pp. 1089–1093, Beijing, China, October 2010.

[38] D. M. Dobkin, *The RF in RFID: Passive UHF RFID in Practice*, Elsevier, San Diego, Calif, USA, 2004.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 394082, 9 pages
http://dx.doi.org/10.1155/2014/394082

*Research Article*

# Pricing Resources in LTE Networks through Multiobjective Optimization

**Yung-Liang Lai[1] and Jehn-Ruey Jiang[2]**

[1] *Department of Computer Science and Information Engineering, Taoyuan Innovation Institute of Technology, Taoyuan, Jhongli 32001, Taiwan*
[2] *Department of Computer Science and Information Engineering, National Central University, Taoyuan, Jhongli 32001, Taiwan*

Correspondence should be addressed to Yung-Liang Lai; yungliang.lai@gmail.com

Received 31 August 2013; Accepted 17 October 2013; Published 2 January 2014

Academic Editors: T. Chen and J. Yang

The LTE technology offers versatile mobile services that use different numbers of resources. This enables operators to provide subscribers or users with differential quality of service (QoS) to boost their satisfaction. On one hand, LTE operators need to price the resources high for maximizing their profits. On the other hand, pricing also needs to consider user satisfaction with allocated resources and prices to avoid "user churn," which means subscribers will unsubscribe services due to dissatisfaction with allocated resources or prices. In this paper, we study the pricing resources with profits and satisfaction optimization (PRPSO) problem in the LTE networks, considering the operator profit and subscribers' satisfaction at the same time. The problem is modelled as nonlinear multiobjective optimization with two optimal objectives: (1) maximizing operator profit and (2) maximizing user satisfaction. We propose to solve the problem based on the framework of the NSGA-II. Simulations are conducted for evaluating the proposed solution.

## 1. Introduction

The 3rd generation partnership project (3GPP) long-term evolution (LTE) technology is one of the major candidates of the fourth generation (4G) wireless communication systems [1]. It offers versatile mobile services using different numbers of resources and enables operators to provide subscribers or users with differential quality of service (QoS) for maximizing subscriber satisfaction. The LTE operators seek the deployment of spectrum-efficient, ubiquitous, always-on, and interoperable mobile broadband wireless access, whose goal is to provide peak data rates of 100 Mbps for high-mobility subscribers and 1 Gbps for low-mobility subscribers [2].

Due to scarcity of resources (e.g., spectrum) in the LTE network, the resources are usually costly. The operators invest huge funds in capital expenditure (CAPEX) and operational expenditure (OPEX) for spectrum licensing and infrastructure construction and management [3], in order to reserve enough resources for subscribers to boost their satisfaction. On one hand, LTE operators need to price the resources high for maximizing their profits. On the other hand, pricing also needs to consider user satisfaction with allocated resources and prices to avoid the "user churn" [4], which means subscribers will unsubscribe services due to dissatisfaction with allocated resources or prices. It is important to study how to price the resources for maximizing the operator profit and maximizing the subscriber satisfaction at the same time in LTE networks.

This paper investigates the pricing resources with profit and satisfaction optimization (PRPSO) problem in the LTE networks to simultaneously maximize the operator profit and subscriber satisfaction. The problem is modelled as a multiobjective problem with two conflicting objectives: (1) maximizing operator profit and (2) maximizing user satisfaction. The PRPSO problem is modeled on the resource block allocation model defined in the 3GPP LTE standard [2]. For an LTE operator, the solutions of the PRPSO problem are helpful for analyzing realistic impacts of investment in spectrum, since the PRPSO problem is formulated on the *resource blocks*, which are the units of allocation mechanism for allocating spectrum resources for subscribers and are the units for providing enough quality of services in the LTE network.

Due to the hardness of solving this problem, we develop a heuristic genetic optimization algorithm, called the PRPSO algorithm, to find the solution based on the nondominated sorting genetic algorithm (NSGA-II) in [5]. For an analyst of pricing resources in LTE networks, the solutions found by the PRPSO algorithm are useful for analyzing the subscribers' satisfaction and the operator's profit, based on subscribers' satisfaction models and costs for acquiring spectrum. The optimal solutions found by the PRPSO algorithm are the Pareto fronts in the multiobjective decision theory. The Pareto fronts are a set of choices that are nondominated by other choices, which are helpful for making tradeoff decisions to achieve two conflicting objectives, subscribers' satisfaction and operator's profits, in LTE networks.

Some optimization studies are proposed for resource pricing, resource reservation, and load balancing in LTE networks. Huang et al. in [6] proposed an adaptive call admission control and resource (or bandwidth) reservation scheme using fuzzy logic control and particle swarm optimization (PSO) for 4G networks. Huang et al. in [7] proposed a resource (or bandwidth) reservation mechanism for neighboring 4G cells based on grey prediction theory and swarm intelligence. Dixit et al. in [8] studied the dynamic pricing problem on maximizing operator revenue in LTE networks. However, the problem of reserving and pricing LTE wireless resources to maximize both the operator profit and subscriber satisfaction is not fully studied.

The rest of this paper is organized as follows. In Section 2, the PRPSO problem is formulated. The proposed heuristic genetic PRPSO algorithm to solve the problem is introduced and evaluated in Sections 3 and 4, respectively. And finally, some concluding remarks are drawn in Section 5.

## 2. Modeling and Problem Formulation

In this section, we introduce the resource allocation model, the user satisfaction model, and the PRPSO problem in LTE networks.

*2.1. Resource Allocation Model in LTE Networks.* The LTE network uses an IP-based network architecture to provide voice and data services. Based on the architecture, the operator can design and sell the products by combining different QoS services. The LTE air interface uses orthogonal frequency division multiplexing (OFDM) with advanced antenna techniques to transmit voice and data simultaneously [2]. The scheduler in the base station (called eNodeB or eNB) is responsible for dynamically scheduling the LTE air interface in both the downlink and uplink directions for subscribers' user equipment (UE). As shown in Figure 1(a), much UE accesses an eNB at the same time, and the eNB needs to provide differential services for the UEs.

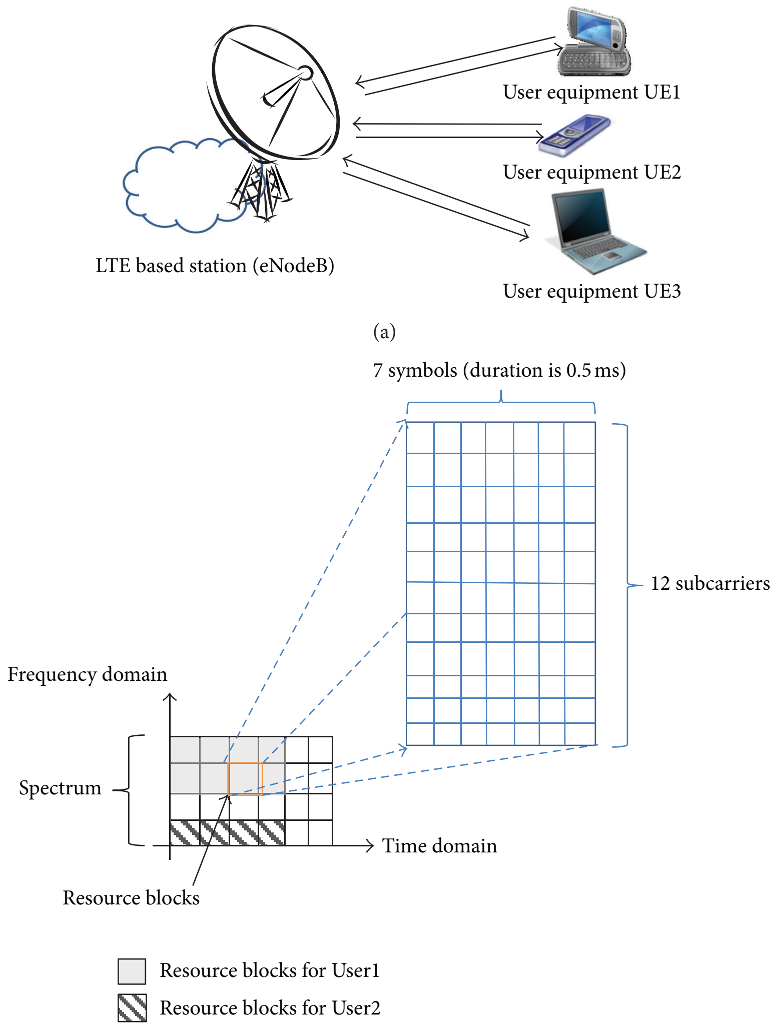
The LTE standard provides a diversity of classes of QoS services [9]. A traffic class or a QoS class is defined according to the restrictions and the limitations of the radio interface. Based on the traffic sensitivity to the packet delay, four classes are defined: the *conversational*, *streaming*, *interactive*, and *background* classes. The conversational class is meant for the traffic that has a high sensitivity to delay (e.g., VoIP), while the background class deals with the traffic that has a low sensitivity to delay (e.g., background downloading files or sending emails). The streaming class is real time based and can preserve time relation (variation) between information entities of streams (say video streams). The interactive class is best effort based and follows a request-response pattern in applications, such as web browsing.

In LTE networks, a set of resource blocks is allocated to a subscriber [10] to provide the services. The more resource blocks are allocated to the subscriber, the better experience of service is, and thus the satisfaction is increased. As shown in Figure 1(b), the downlink (eNB to UE) and uplink (UE to eNB) in the air interface are divided into a number of 15 kHz subchannels in the frequency domain and a number of 0.5 ms time slots in the time domain. The resource block (RB) is the main unit used to schedule transmissions over the air interface (refer to Figure 1(b)). An RB contains 12 contiguous subchannels and 7 symbols (duration is 0.5 ms). In general, a number of RBs are allocated to an UE according to its quality of service (QoS) [11]. Statistically, the more the resource blocks are allocated to the UE of a subscriber, the more the subscriber is satisfied with the service. In Figure 1(b), the number of allocated resource blocks of User 1 is higher than that of User 2, which implies that the satisfaction of the User 2 is higher than User 1.

The resource block based model is more accurate in analysing the channel resources used in the LTE network, since it considers not only the cost of transmitting data but also the overhead cost (such as retransmissions when packets are corrupted). It is notable that most of studies on pricing are based on the received usage based model to charge subscribers based on the amount of received packets, which do not include the overhead cost.

*2.2. User Satisfaction Model.* An operator allocates or reserves resources to subscribers to provide them with differential QoS levels. The satisfaction of subscribers is important to the operator. Without sufficient resources allocated to the services, subscribers will feel dissatisfied. For example, the acceptable one-way (speaker's mouth to listener's ear) delay of voice communication for VoIP applications recommended by ITU [12] is at most 150 ms. The subscriber of VoIP service will feel dissatisfied if transmission delay is more than 150 ms due to the insufficiency of allocated resources. Dissatisfied subscribers may unsubscribe some services, which will damage the operator's profits. They may unsubscribe all services and migrate to another operator, causing the "user churn" problem reported in [4].

Lin et al. in [13] proposed a method to approximate the subscriber's satisfaction with the allocated resources by a sigmoid function. In this paper, we also adopt the sigmoid function to model the user satisfaction. The sigmoid function is useful for modeling natural processes or system learning curves, since it can represent a history dependent progression



(a)

(b)

Figure 1: Illustration of services controlled and resources allocated by an LTE based station (eNodeB) in the LTE network. (a) UE with differential QoS requirements of services and (b) resource blocks allocated for a UE by a eNodeB in the frequency domain and time domain.

approaching a limit. It depends on a random variable $x$ to represent the occupation of resources for the subscriber. The sigmoid function is formulated as follows:

$$\Psi(x) = \frac{1}{\left(1 + e^{-\alpha(x-\beta)}\right)}, \qquad (1)$$

where $\alpha$ and $\beta$ are the steepness and the middle of the curve, respectively.

Figure 2 plots curves of the sigmoid functions, where $\alpha$ stands for the sensitivity and $\beta$ is the median value of the satisfaction curve. As shown in Figure 2(a), the curve is with higher steepness as $\alpha$ is higher. As shown in Figure 2(b), the starting point of the curve is farther away from zero as $\beta$ is higher.

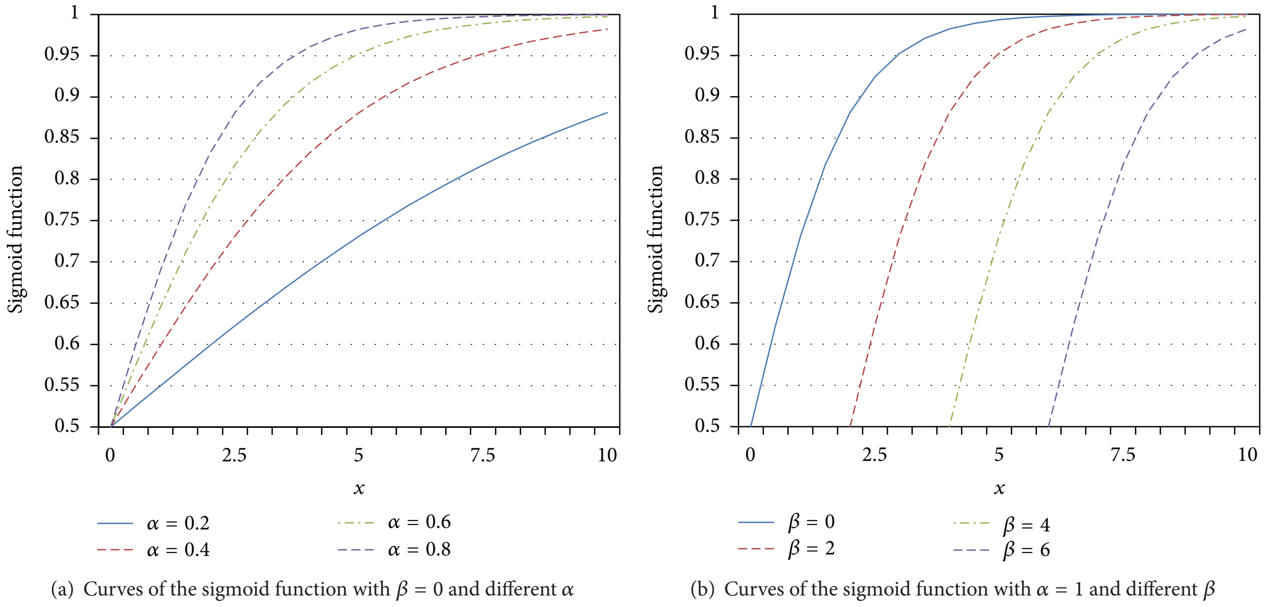
In subscriber's point of view, the value of $\alpha$ indicates the subscriber's sensitivity to the degradation of service, while $\beta$ indicates the acceptable level for the service. It is remarkable that $\beta$ decides when the satisfaction starts to increase and $\alpha$ decides how fast the satisfaction increases.

### 2.3. *The Pricing Resources with Profit and Satisfaction Optimization (PRPSO) Problem.*

We formulate the pricing resources with profit and satisfaction optimization (PRPSO) problem in this subsection. The main goals of the problem are (1) to maximize the operator's profit and (2) to maximize the users' satisfaction. The formulation is based on a fixed period of time $T$, say 1 day, 1 month, 2 months, 1 year, and so on.

The first goal is to maximize the operator's profit. Equation (2) is used to formulate the operator's profit, $P$, which consists of two factors, revenue from subscribers (RS) and cost of spectrum (CS). Below we explain the meaning of (2) and describe some assumptions and notations used in it.

We assume an operator has a set of spectrum segments (notated by $\Phi$). For segment $i \in \Phi$, $B_i$ represents the quantity of units occupied in segment $i$, and $C_i$ represents the cost per unit of segment $i$. Thus, the operator pays $\text{CS} = \sum_{i\in\Phi} B_i C_i$ for acquiring the spectrums for period $T$. We also assume the operator sells a set of services (notated by $\Omega$) to subscribers. By statistics or by predictions, the operator totally allocates $Q_s$



(a) Curves of the sigmoid function with $\beta = 0$ and different $\alpha$

(b) Curves of the sigmoid function with $\alpha = 1$ and different $\beta$

Figure 2: Curves of the sigmoid functions.

resource blocks for each service $s \in \Omega$ over period $T$. Therefore, if the subscribers pay the price $P_s$ for each resource block allocated to service $s$ for period $T$, the operator has revenue RS $= \sum_{s\in\Omega} Q_s P_s$ from the subscribers. For example, assume a service $Sa$ is averagely allocated 100 resource blocks per day and the price of each resource block allocated to service $Sa$ is 1.2 dollar. It means $Q_{sa}$ is 100 resource blocks and $P_{sa}$ is 1.2 dollar per day; thus the revenue is 120 dollar per day for service $Sa$. Consider

$$P = \mathrm{RS} - \mathrm{CS} = \sum_{s\in\Omega} Q_s P_s - \sum_{i\in\Phi} B_i C_i. \tag{2}$$

$P_s$ denotes the price per resource block allocated to service $s$. In reality, $P_s$ is a bounded price variable and $P_s \in [P_{\min}, P_{\max}]$ as shown in

$$P_{\min} \leq P_s \leq P_{\max}. \tag{3}$$

The second goal is to maximize the subscribers' satisfaction. Equation (4) is used to formulate the satisfaction per paid price $U$. In (4), $\psi_s(Q_s)$ is used to formulate the satisfaction for service $s$; it is a sigmoid function of $Q_s$, the number of resource blocks allocated to service $s$. Note that $Q_s P_s$ is the price a subscriber pays for using service $s$. Therefore, $\psi_s\ (Q_s)/Q_s P_s$ is the satisfaction per unit of paid price for service $s$. $U$ is hence the overall subscribers' satisfaction per unit of paid price

$$U = \sum_{s\in\Omega} \psi_s \frac{(Q_s)}{Q_s P_s}. \tag{4}$$

Given $Q_s$, $B_i$, $C_i$, and $\psi_s(Q_s)$, $\forall\ i \in \Phi$, $s \in \Omega$, the PRPSO problem is to find a price set PS for maximizing both the operator's profit and subscribers' satisfaction, defined as

$$\begin{gathered} \text{Maximize } P, \\ \text{Maximize } U. \end{gathered} \tag{5}$$

Now, we discuss some issues of estimating parameters in the PRPSO problem. The PRPSO problem is an optimization problem to decide prices, based on given information. The quantity $Q_s$ of resource blocks allocated to service $s$ is possibly estimated from historic usage of resource blocks allocated to service $s$ over the fixed period of time $T$. The per-unit cost $C_i$ of spectrum segment $i$ is also possibly estimated as the average cost of acquiring and managing spectrum over the time period $T$. When the solutions of PRSP problem are found, the output prices are also on the basis of time period $T$. For example, if $Q_s$ and $C_i$ are estimated over the period of one month, the prices are on the basis of one month. It is also notable that the parameters $\alpha$ and $\beta$ of $\psi_s$ can be adjusted according to subscribers' experiences to shape the sigmoid function properly.

## 3. The Pricing Resources with Profit and Satisfaction Optimization (PRPSO) Algorithm

In this section, we present our multiobjective pricing algorithm, called the PRPSO algorithm, to solve the pricing resources with profit and satisfaction optimization (PRPSO) problem. The proposed PRPSO algorithm is based on an evolutionary genetic algorithm (GA) approach, which is used to heuristically find the solutions of optimization problems. The GA approach is to mimic natural selection in the biology, where individuals with higher fitness can survive to next generation [14].

In the GA approach, the population (a set of individuals or solutions) is randomly generated in the initial step. Then, the population evolves in the generation loop for MAX_GEN times. In each generation, fundamental operations, such as selection, crossover, and mutation are used to generate individuals into the next generation. When the generation loop is terminated, the solution is made by decoding the best individuals in the decode step.

Based on the above steps and based on the nondominated sorting genetic algorithm II (NSGA-II) algorithm in [5], we design the PRPSO algorithm for finding good solutions to the PRPSO problem. The basic idea of the NSGA-II algorithm is to find, from the solutions of the current and the next generations, the optimal front (called Pareto front), which is the set of nondominated feasible solutions (or front points) that are not dominated by any others. It is noted that a solution $x$ is said to dominate another solution $y$, if and only if $x$ is better than $y$ in at least one evaluation of objectives and $x$ is not worse than $y$ in all evaluations of objectives.

$N$ solutions in the Pareto front are selected to evolve, as the population is assumed to be of size $N$ for each generation. If the first-found optimal front (or call the first optimal front) has less than $N$ front points, then the second optimal front should also be found. The second optimal front is the set of nondominated feasible solutions over all population members except for those in the first optimal front. If the first and the second optimal fronts totally have less than $N$ members, then the third optimal front should be found further and so on. Not all the front points in the last-found optimal front are selected. They are in practice selected according to the fitness (i.e., the nondomination) and the spread of solutions so that the optimal front found in the final generation will have better convergence near the true Pareto front. It is noted that the notion of crowding distance is used for evaluating the degree of spread of solutions.

Now, we introduce how to evaluate an individual of a population in the proposed algorithm. Each individual (say $x$) in the population has two attributes: (1) nondomination $x_{\text{rank}}$ and (2) crowding distance ($x_{c\_\text{dist}}$), where $x$ has rank 1 (or $2, 3, \ldots$) if it belongs to the 1st (2nd, 3rd, ...) optimal front, and the crowding distance is the summation of distances between $x$ and two adjacent individuals in every evaluation of objectives (please refer to [5] for the details of crowding distance calculation). A partial order $\prec_n$ is defined between two individuals $x$ and $y$ in

$$x \prec_n y, \quad \text{if} \begin{cases} x_{\text{rank}} < y_{\text{rank}} \\ x_{\text{rank}} = y_{\text{rank}}, x_{c\_\text{dist}} > y_{c\_\text{dist}}. \end{cases} \tag{6}$$

In (6), between two individuals or solutions with differing nondomination ranks, we prefer the solution with the lower (better) rank. Otherwise, if both solutions belong to the same front, then we prefer the solution that is located in a less crowded region.

The PRPSO algorithm runs generation by generation. In each generation, a front set $F = \{F_1, F_2, \ldots, F_r\}$ is produced from both populations of the current and the previous generations, where $F_1, F_2, \ldots, F_r$ are the 1st, 2nd, ..., $r$th optimal fronts and $r$ is the maximum number of fronts to be accommodate in a population of size $N$ (i.e., $|F_1| + |F_2| + \cdots |F_{r-1}| < N$ and $|F_1| + |F_2| + \cdots |F_{r-1}| \geq N$). As shown in the example in Figure 3, there are three fronts ($F_1$, $F_2$, and $F_3$) produced on the two dimensional space, where the two dimensions correspond to the two objective functions OF1 and OF2. Front $F_1$ is the set of solutions that are not dominated by any others. Each solution in front $F_i$ is not dominated by any solution in front $F_j$, for all $j > i \geq 1$. The optimization goals in the PRPSO problem are to maximize the OF1 (i.e., profit: $\mathbf{P}$, defined in (3)) and OF2 (i.e., satisfaction: $\mathbf{U}$, defined in (4)), so an optimal front is the farthest from the origin point.

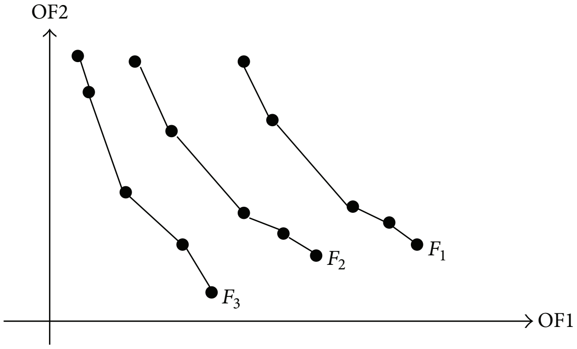


Figure 3: Illustration of a front set $F$, where $F = \{F_1, F_2, F_3\}$. Each point represents one feasible solution in one front in the 2-dimensional space. $F_1$ (resp., $F_3$) is best (resp., worst) front, and the solutions in $F_1$, $F_2$, and $F_3$ have the nondomination rank of 1, 2, and 3, respectively.

Since the populations are generated from the parents with the best finesses of the previous generation, the goodness of populations will be improved after some generations. In addition, the diversity of solutions is kept by the crowding distance so that the solutions widely spread. In this way, when the algorithm terminates, the returned optimal front $F_1$ will be very close to the real Pareto front.

The pseudocode of the PRPSO algorithm is shown in Algorithm 1. Initially, the generation counter $t$ is 0 and the population $P_t$ is randomly generated, where a member in $P_t$ is an individual (or a solution) consisting of the price variable, which is a vector for multiple service cases. An offspring population $Q_t$ is set as empty initially.

As illustrated in Algorithm 1, in step S1, we set $H_t$ to be the union of $P_t$ and $Q_t$. The step S1 is also illustrated in Figure 4. In step S2, the algorithm evokes the Nondominated_Fronts_Sort ($H_t, \mathbf{P}, \mathbf{U}$) subroutine to sort solutions according to their nondomination ranks to have a front set $F = \{F_1, F_2, \ldots, F_r\}$.

The step S3 is to set the population $P_{t+1}$ to be empty and set the counter $i$ to be 1 before the algorithm enters the loop in step S4. The step S4 is to insert the nondominated solutions into $P_{t+1}$. The step S5 is to generate a sorted $F_i$ by the crowding distance in the descending order. The step S6 is to insert the most widely spread ($N - |P_{t+1}|$) solutions using the crowding distance values in the sorted front $F_i$ into the $P_{t+1}$.

The step S7 is to create new offspring population $Q_{t+1}$ from $P_{t+1}$ by mutation and crossover operations, where the

```
Input:
**P**: Profit function
**U**: Utilization function
Output: Pareto front
// Initialization
$t = 0$; $P_t$ = random population; $Q_t = \emptyset$
// Main Loop
S1: $H_t = P_t \cup Q_t$
S2: $F$ = Nondominated_Fronts_Sort($H_t$, **P**, **U**), where $F = \{F_1, F_2, \ldots, F_r\}$ is the front set
S3: $P_{t+1} = \emptyset$; $i = 1$
S4: While ($|P_{i+1}| + |F_i| < N$) Do {$P_{t+1} = P_{t+1} \cup F_i; i + +;$}
S5: Crowding_Distance_Sort($F_i$)
S6: Insert the first ($N - |P_{t+1}|$) elements in the sorted $F_i$ into $P_{t+1}$
S7: $Q_{t+1} \leftarrow$ GenerateNewPouluation($P_{t+1}$)
S8: If ($t <$ MAX_GEN) Then {$t = t + 1$; Goto S1; } Else Return the Pareto front $F_1$
```

Algorithm 1: Pricing resources with profit and satisfaction optimization (PRPSO) algorithm.

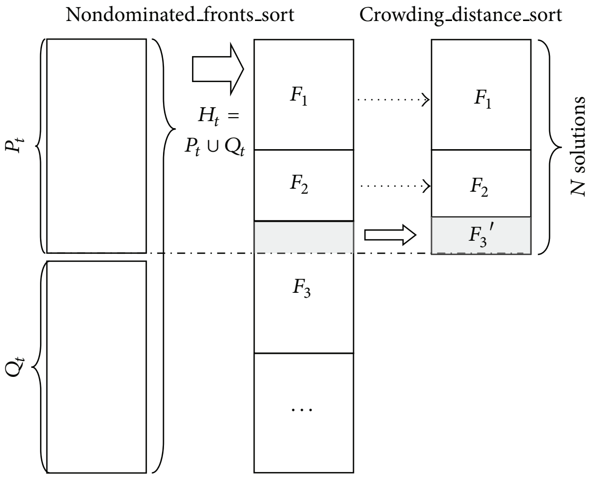


Figure 4: Procedures of generating new population $P_{t+1}$ from $P_t$ and $Q_t$, where $P_t$ is the parent population and $Q_t$ is the child population.

size of $Q_{t+1}$ is $N$. In step S8, the algorithm checks whether the maximum generation is reached. If the generation counter $t$ is less than the maximum value (MAX_GEN), then $t$ is increased by 1 and then the algorithm goes to step S1; otherwise, the algorithm terminates.

## 4. Evaluation

In this section, we evaluate the effectiveness of proposed algorithm. The simulation is conducted by the simulator developed on Matlab [15]. The simulation of the proposed algorithm is conducted with following setting in Table 1.

The parameters used in the simulation are listed as follows. The initial population is created using a uniform random distribution. The population size is $15 \cdot |X|$, where $|X|$ is the number of prices, each of which corresponds to a service. The price is a real number, whose range is from 1 to 2. We set the Pareto fraction as 0.35, which means the algorithm will try to limit the number of individuals in the current population that are on the Pareto front to 35 percent of the population size.

Table 1: Parameter setting.

| Parameter | Values |
|---|---|
| Number of prices (services) | $|X|$, where $|X|$ is $3, 5, \ldots, 17$ |
| Initial population | Uniform random distribution |
| Population size | $15 \cdot |X|$ |
| Range of price variable ($X$) | (1, 2) |
| Pareto fraction | 0.35 |
| StallGenLimit | 100 |
| Toleratethreshold | $1 \times 10^{-6}$ |
| MAX_GEN | $200 \cdot |X|$ |

In the simulation, two conditions are used to determine whether to stop the algorithm execution. In Condition-1, the algorithm stops when the maximum number of generations (MAX_GEN) is reached, where the MAX_GEN is $200 \cdot |X|$. In Condition-2, the algorithm stops if the average change in the spread of the Pareto front over the "StallGenLimit" generations is less than the tolerable threshold (TolerateThresold). The algorithm stops when either of the conditions is satisfied.

*4.1. Evaluation of Tradeoff Relationship of Two Conflicting Objectives.* We first simulate the proposed algorithm in the basic setting, which is to decide price variables for three services, for the evaluation of tradeoff relationship of two conflicting objectives. The simulation results are shown in Figure 5, where Objective 1 is operator's profit and Objective 2 is subscribers' satisfaction per unit of paid price. Several front points are plotted in Figure 5, which form the Pareto front of the multiobjective optimization theory. Each point has two values, which are the operator's profit and the subscriber's satisfaction. As shown in Figure 5, the lower profit implies the higher satisfaction, while the higher profit implies the lower satisfaction. In summary, it is impossible to increase the profit and satisfaction at the same time, and thus there is tradeoff

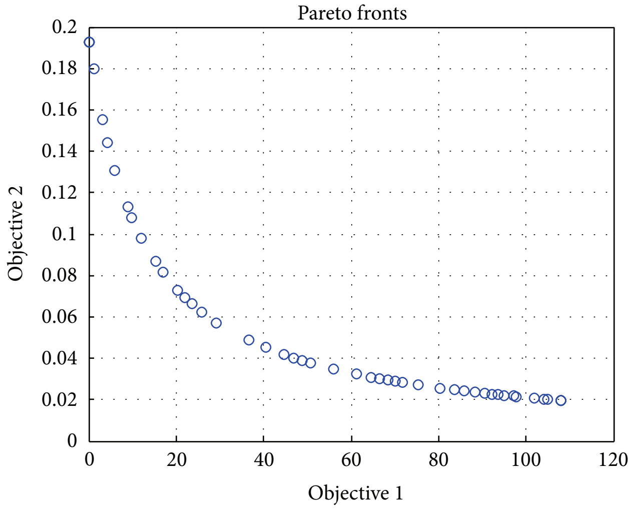


Figure 5: Results of pricing under two conflicting objectives, where Objective 1 is the operator's profit and Objective 2 is subscribers' satisfaction.

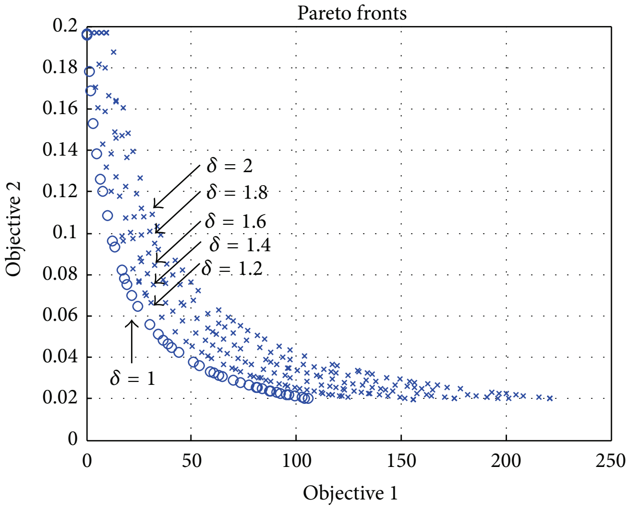


Figure 6: Evaluation of the effect of raising prices by adjusting the price raising factor $\delta$.

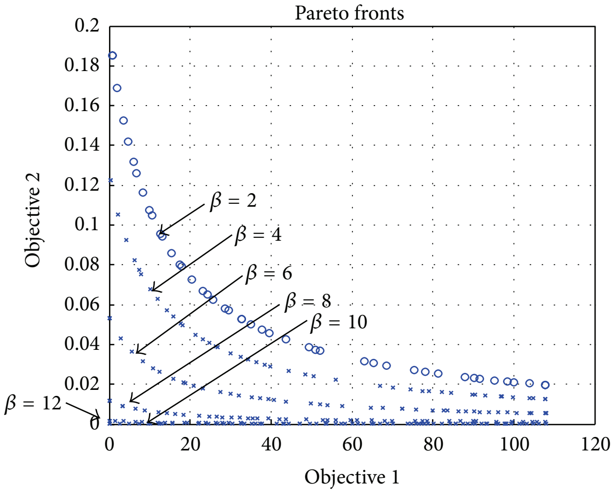


Figure 7: Evaluation of the effect of changing sensitivity of satisfaction on services.

between the two objectives: the operator's profits and subscribers' satisfaction.

*4.2. Evaluation of Impacts of Raising Prices of Two Conflicting Objectives.* In this section, we study the effectiveness of raising prices of the services. We add into (2) an additional variable for controlling price raising factor $\delta$ to have (7), where the price raising factor $\delta$ is $1, 1.2, \ldots, 2$,

$$P = \sum_{s\in\Omega} Q_s\left(\delta \cdot P_s\right) - \sum_{i\in B} B_i C_i. \tag{7}$$

As shown in Figure 6, the maximum values of profit (Objective 1) of Pareto fronts move to the right, if the price raising factor $\delta$ is increased. It reflects the effectiveness of raising prices to increase the profit.

The solutions found by the proposed algorithm are stable, since the results do not fluctuate along the curves, as shown in Figure 6. Moreover, the effect of raising prices can be easily observed in the results. For example, the maximal profit of original curve ($\delta = 1$) is 108, and the maximal profit of adjusted curve ($\delta = 2$) is 214, as shown in the Figure 6. Therefore, the maximum profit is almost doubled, meaning the effect of raising price is obvious.

*4.3. Evaluation of Impacts of Raising Prices of Two Conflicting Objectives.* We study in this section the effect of changing of the median value ($\beta$) of the satisfaction of services. We set the median value ($\beta$) as $2, 4, \ldots, 12$ in order to analyze the corresponding results. As shown in Figure 7, if $\beta$ is increased, satisfaction is decreased. This is because a subscriber starts to feel satisfied only after a lot of resources are allocated to him/her for the cases of higher $\beta$ values.

The results show that the difference of satisfactions for different subscriber types is obvious by the results found by the PRPSO algorithm. Moreover, the characteristics of relationship of operator profit and subscriber satisfaction can be easily observed based on the results. For example, in the $\beta = 12$ case, the satisfaction is almost the same, even if the profit reaches the maximum value.

*4.4. Evaluation of Efficiency of Finding Pareto Fronts.* Fourth, we evaluate the quality metrics of forming the Praetor front in different number of decision variables. The quality metrics are (1) the average distance of Pareto front and (2) number of points of Pareto front. In general, a smaller average distance indicates that the solutions on the Pareto front are evenly distributed. The average distance is the crowding distance, which is the perimeter of the cuboid formed by using the nearest neighbors as the vertices in the Pareto front; please refer to the paper [5] for more details. The number of points of Pareto front indicates the tractability of the Pareto front for a decision maker. When the number of points or solutions of Pareto

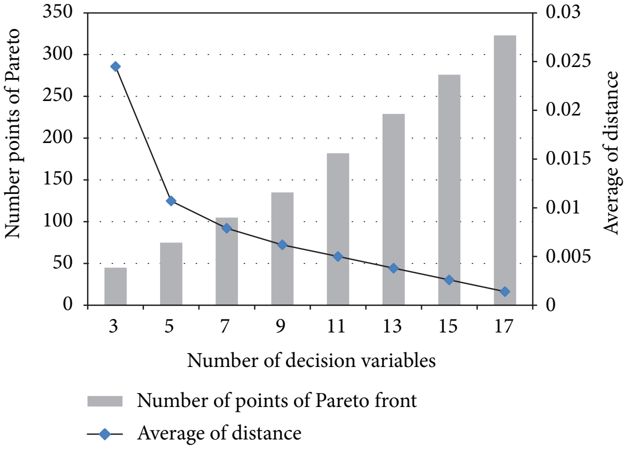


Figure 8: Evaluation of the Pareto front in terms of (1) the number of points of Pareto front and (2) the average of distance, where the right-side $y$-axis is the average distance of the Pareto front, the left-side $y$-axis is the number of points of the Pareto front, and the $x$-axis is the number of decision variables, which is the number of prices (or services).

front are too large, then the solutions may be intractable for a decision maker.

As shown in Figure 8, the number of points of the Pareto front is increased, but the average distance is decreased, when the number of decision variables is increased. It implies that more points are included in the Pareto front, when the number of decision variables is larger. Selecting a pricing solution from a larger set is more intractable for a decision maker facing higher numbers of price variables. Hence, the decision maker needs to carefully make decisions when they face a higher number of price variables.

## 5. Conclusions

The operators invest huge funds for acquiring the spectrum resources in the LTE network. The operator profit and the subscriber satisfaction are two most important factors. Thus, it is necessary to consider the operator profit factor and subscriber satisfaction factor for pricing resources in the LTE networks. However, most of existing studies only consider the problem about maximizing operator profit. This paper investigates the pricing resources with profit and satisfaction optimization (PRPSO) problem in the LTE network to simultaneously maximize the operator profit and subscriber satisfaction. This paper contributes a theoretical framework to help decision makers in pricing resources, based on the heuristic optimization algorithm—PRPSO algorithm. Compared with the algorithm only solving a single pricing optimization goal, the PRPSO algorithm solves the optimal problem with the consideration of two important goals, which is more helpful for making decisions in pricing.

The PROSO algorithm has been verified and tested by the simulations on the basis of convergence and diversity performance metrics to guarantee the quality of optimal solutions found. The simulation results show that the difference of satisfactions for different subscriber types is obvious. Moreover, the characteristics of relationship of the operator profit and the subscriber satisfaction can also be easily observed based on the results.

## Conflict of Interests

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

## References

[1] F. Beltrán, J. A. Gutiérrez, and J. L. Melús, "Technology and market conditions toward a new competitive landscape in the wireless access market," *IEEE Communications Magazine*, vol. 48, no. 6, pp. 46–52, 2010.

[2] D. Astély, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall, "LTE: the evolution of mobile broadband," *IEEE Communications Magazine*, vol. 47, no. 4, pp. 44–51, 2009.

[3] P. Bhat, S. Nagata, L. Campoy et al., "LTE-advanced: an operator perspective," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 104–114, 2012.

[4] K. Coussement and D. Van den Poel, "Churn prediction in subscription services: an application of support vector machines while comparing two parameter-selection techniques," *Expert Systems with Applications*, vol. 34, no. 1, pp. 313–327, 2008.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[6] C.-J. Huang, Y.-T. Chuang, and D.-X. Yang, "Implementation of call admission control scheme in next generation mobile communication networks using particle swarm optimization and fuzzy logic systems," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1246–1251, 2008.

[7] C.-J. Huang, H.-Y. Shen, and Y.-T. Chuang, "An adaptive bandwidth reservation scheme for 4G cellular networks using flexible 2-tier cell structure," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6414–6420, 2010.

[8] S. Dixit, S. Periyalwar, and H. Yanikomeroglu, "Secondary user access in LTE architecture based on a base station centric framework with dynamic pricing," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 284–296.

[9] H. Ekström, "QoS control in the 3GPP evolved packet system," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 76–83, 2009.

[10] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: next-generation wireless broadband technology," *IEEE Wireless Communications*, vol. 17, no. 3, pp. 10–22, 2010.

[11] B. Sadiq, R. Madan, and A. Sampath, "Downlink scheduling for multiclass traffic in LTE," *Eurasip Journal on Wireless Communications and Networking*, vol. 2009, Article ID 510617, 18 pages, 2009.

[12] "ITU-T 1996, one-way transmission time," ITU-T Recommendation G.114, 1996.

[13] H. Lin, M. Chatterjee, S. K. Das, and K. Basu, "ARC: an integrated admission and rate control framework for CDMA data networks based on non-cooperative games," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 326–338, ACM, September 2003.

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

[15] Mathworks Corporation, Matlab, http://www.mathworks.com.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 418396, 9 pages
http://dx.doi.org/10.1155/2013/418396

*Research Article*

# Comparison of Multiobjective Evolutionary Algorithms for Operations Scheduling under Machine Availability Constraints

**M. Frutos,[1] M. Méndez,[2] F. Tohmé,[3] and D. Broz[1]**

[1] *Department of Engineering and Instituto de Investigaciones Económicas y Sociales del Sur (IIESS-CONICET), Universidad Nacional del Sur, Avenida. Alem 1253, 8000 Bahía Blanca, Argentina*
[2] *Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI), Universidad de Las Palmas de Gran Canaria (ULPGC), Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain*
[3] *Department of Economics and Instituto de Investigaciones Económicas y Sociales del Sur (IIESS-CONICET), Universidad Nacional del Sur (UNS), 12 de Octubre 1198, 8000 Bahía Blanca, Argentina*

Correspondence should be addressed to M. Frutos; mfrutos@uns.edu.ar

Received 30 September 2013; Accepted 5 December 2013

Academic Editors: T. Chen and Q. Cheng

Many of the problems that arise in production systems can be handled with multiobjective techniques. One of those problems is that of scheduling operations subject to constraints on the availability of machines and buffer capacity. In this paper we analyze different Evolutionary multiobjective Algorithms (MOEAs) for this kind of problems. We consider an experimental framework in which we schedule production operations for four real world Job-Shop contexts using three algorithms, NSGAII, SPEA2, and IBEA. Using two performance indexes, Hypervolume and R2, we found that SPEA2 and IBEA are the most efficient for the tasks at hand. On the other hand IBEA seems to be a better choice of tool since it yields more solutions in the approximate Pareto frontier.

## 1. Introduction

Production planning problems involve the allocation of scarce resources to different tasks in such way as to optimize one or more efficiency-related goals [1]. In most cases, these problems are analyzed as instances of the Job-Shop Scheduling Problem, in which given a set of machines and a list of jobs, represented as ordered sequences of operations, to be run on the machines, the goal is to minimize, in particular, the processing time of all jobs, known as *makespan* [2]. This problem belongs to the class NP-hard [3], for which no efficient algorithms are known to run in reasonable execution times. The literature focuses mostly on single-objective versions of the problem, despite the fact that several authors have stated that a genuine scheduling problem involves more than one objective when production efficiency is sought [4]. If this is indeed the appropriate approach, Multiobjective Evolutionary Algorithms (MOEAs) are the tools of choice [5, 6]. Their main advantages are their ease of adaptation to different instances and their overall efficiency. Only the fitness function has to be known, instead of its rates of variation, making the evolutionary algorithms efficient for problems that cannot be solved in reasonable time with gradient-based methods.

In this paper we will evaluate the performance of three MOEAs on Job-Shop scheduling problems [7, 8]: NSGAII [9], SPEA2 [10], and IBEA [11]. These competing approaches all use domination and elitism to reach the best possible approximation to the solutions of multiobjective problems. They differ in the strategies on which they are based (lack of a predefined sharing parameter, in the case of NSGAII; a fine grained fitness assignment procedure in SPEA2; a qualitative indicator function over Pareto approximations in IBEA), but their performance on different instances tends to be the best available in the literature. We, furthermore, add extra constraints, on both the availability of machines and the buffer capacity. We run the algorithms on real-world problems in which nonstandardized production (like in the Job-Shop context) has to share machines with standardized processes that have priority over the former.

## 2. Job-Shop Scheduling Problem

The Job-Shop Scheduling Problem is quite complex. It has been analytically solved for 1, 2, and 3 machines and a small number of jobs. Only a few efficient algorithms have been found for 4 or more machines and 3 or more jobs. This is due to the combinatorial explosion of possible schedules. In the next subsections we will review the state of the art in this matter and formally define the problem, introducing the objectives to be optimized in our analysis.

*2.1. State of the Art.* A brief review of the approaches to the Job-Shop Scheduling Problem shows a multiplicity of techniques. So, for instance, [12] presents a tabu search method intended to minimize total tardiness, while [13] presents a GRASP (Greedy Randomized Adaptive Search Procedures) algorithm that minimizes the makespan, and [14] uses a HACO (Hybrid Ant Colony Optimization) algorithm for the same goal. In [15] a localization approach is suggested, minimizing both makespan and the machine load. In [16] a mathematical model is introduced, able to solve only small instances of the problem. Closer to our object of study, [17] presents a hybrid algorithm also minimizing makespan, while [7] introduces a genetic algorithm in which the representation makes every schedule feasible. In [18] a genetic algorithm is presented for which the control parameters have been tuned to optimize makespan. In [19] dispatch rules are proposed and at each generation the search space is explored by means of schemes, again with the objective of minimizing makespan. This approach is generalized in [20] where an architecture based on an evolutionary algorithm is combined with learning through schemes and the generation of populations by means of combined dispatch rules. In [21] a multiple scenarios genetic algorithm is introduced, in which each scenario corresponds to an operation and each feasible machine to a state. In [22] a genetic algorithm profits from the localization approach presented in [15]. A class of mutations consists in allocating operations from machines with heavy loads to less loaded ones. In [2] a hybrid genetic algorithm solves the problem with the proviso that no waiting time is allowed among operations for any job, minimizing the makespan. Finally, [23] presents an evolutionary algorithm minimizing the makespan, the total work load, and the maximum load.

*2.2. Formal Definitions.* The Job-Shop Scheduling problem consists in finding an optimal allocation of a class of n jobs $J = \{J_j\}_{j=1}^{n}$, to be processed by a set of m machines $M = \{M_k\}_{k=1}^{m}$. Each job is described as a sequence of tasks that can be performed in sequence $J_j \equiv S_1, \ldots, S_{nj}$. The operation of a task $S_i$ of job $J_j$ on machine $M_k$ is denoted as $O_{ij}^{k}$. Operation $O_{ij}^{k}$ requires using machine $M_k$ for an uninterrupted *processing time* $\tau_{ij}^{k}$. A solution for this problem involves the determination of the starting time $t_{ij}^{k}$ of each operation $O_{ij}^{k}$ while optimizing the objectives [8]. Here we consider three objectives. The first one is the minimization of the *makespan* (1). This involves shortening the total time required for the $n$ jobs. The second objective is the minimization of the *mean flow time* (2). This amounts to reducing the number of jobs processed in parallel. Finally, we seek to minimize the effects on the makespan of variations of the $\tau_{ij}^{k}$, for each $O_{ij}^{k}$. For this we run microsimulations to find the *variance* of the first objective $\sigma^2(C_{\max}^{j})$ [24]. The minimization of this variance ensures the stability of solutions (3):

$$f_1 : \min C_{\max}^{j} = \sum_{i \in S_j} \max_{k \in M} \left( t_{ij}^{k} + \tau_{ij}^{k} \right), \tag{1}$$

$$f_2 : \min \overline{F} = \frac{1}{n} \sum_{j \in J} \sum_{i=m} \left( t_{ij}^{h} + \tau_{ij}^{h} - t_{1j}^{k} \right), \tag{2}$$

$$f_3 : \min \sigma^2 \left( C_{\max}^{j} \right). \tag{3}$$

We assume the nonnegativity of the starting time $t_{ij}^{k}$ of each $O_{ij}^{k}$: $t_{ij}^{k} \geq 0$. Besides, we have (joint) precedence constraints of operations for each job: if $i \geq s$, $S_i$, $S_s$ are tasks of $J_j$ and $S_i$ is executed on $M_k$, while $S_s$ on $M_h$, then $t_{ij}^{k} - t_{sj}^{h} \geq \tau_{sj}^{h}$. Finally, the (disjoint) nonjuxtaposition constraints are applied on each machine: if $S_i$ is a task in $J_j$ while $S_s$ in $J_p$, both to be executed on $M_k$, we have that $t_{ij}^{k} - t_{sp}^{k} \geq \tau_{sp}^{k}$. The purpose of the latter constraints is to warrant that no machine carries out two operations at the same time. Two additional constraints involve the availability of machines and the capacity of the buffer. The first ones limit the operational interval of each machine; that is, $(t_{ij}^{k}, t_{ij}^{k} + \tau_{ij}^{k})$ must be larger than the operational interval corresponding to the standardized operational interval of machine $M_k$. The second group of constraints limits the number of operations $O_{ij}^{k}$ on machine $M_k$ that are on a waiting list. The buffer can either hold 0 operations (*no-wait*) or $n - 1$ operations (*nonrestricted*).

## 3. Evolutionary Multiobjective Algorithms

Evolutionary algorithms imitate genetic processes by improving solutions, pairing existing solutions as if they were DNA chains, and creating new chains. A *chromosome* is composed by smaller units called *genes*. For our problem the chromosomes identify a schedule of operations. Evolutionary improvements should end up yielding the optimal schedule. To show how this works we use an example with three jobs and three machines (3 × 3). The total problem involves nine operations. Their corresponding processing times ($\tau_{ij}^{k}$) and variances ($\sigma_{ij}^{k^2}$) for the machine in which they run ($M_k$) are shown in Table 1.

*3.1. The Evolutionary Phase.* To represent an individual schedule, we use the notation proposed in [8]. The chromosome contains binary variables and the chain has as many genes as machines in the problem. Each gene has a certain number of *alleles*, depending on the number of jobs of the problem. More precisely, the size of a gene is $\lceil \log(n!)/\log(2) \rceil$, while the total size of the chromosome is $m * \lceil \log(n!)/\log(2) \rceil$. For each gene the sequence of binary numbers represents the sequence of jobs in the corresponding machine $M_k$. In our

Table 1: Scheduling operations.

| 3 × 3 problem with 9 operations | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $J_j$ | | $J_1$ | | | $J_2$ | | | $J_3$ | |
| $O_{ij}^k$ | $O_{11}^1$ | $O_{21}^2$ | $O_{31}^3$ | $O_{12}^2$ | $O_{22}^3$ | $O_{32}^1$ | $O_{13}^3$ | $O_{23}^1$ | $O_{33}^2$ |
| $\tau_{ij}^k$ | 10 | 15 | 8 | 12 | 9 | 10 | 11 | 5 | 16 |
| ${\sigma_{ij}^k}^2$ | 4 | 4 | 2 | 4 | 3 | 5 | 2 | 1 | 4 |
| $M_k$ | 1 | 2 | 3 | 2 | 3 | 1 | 3 | 1 | 2 |

example we choose 000 → 1 | 2 | 3, 001 → 1 | 3 | 2, 010 → 2 | 1 | 3, 011 → 2 | 3 | 1, 100 → 3 | 1 | 2 and 101 → 3 | 2 | 1. Consider the first parent in Figure 1. The first gene, of machine $M_1$, is 000. This means that the sequence of jobs in the machine is 1, 2, and 3. For $M_2$ the gene is 010, that is, the sequence of jobs, is 2, 1, and 3. Finally, for $M_3$ the gene is 011, and therefore the jobs are 2, 3, and 1.

A crossover operator acts on pairs of chromosomes. It aligns the chromosomes, cuts them at a certain points, and exchanges the fragments between the chromosomes. To see how it works, consider again Figure 1 the two "parents" to crossover, called First Parent and Second Parent. A "child" is built incorporating randomly elements from both parents (the offspring in Figure 2). The other child is obtained by inverting the choices made for the other one. This crossover, called uniform, yields better results exploring solutions close to the Pareto frontier. A mutation varies the binary values of one or more alleles of the gene. This variation is applied at random points of the chromosome, generating an individual with small differences with the original chromosome. In our case 10% of the alleles of the chromosomes are changed. In Figure 2, offspring* represents the mutated chromosome.

*3.2. Selection of MOEAs.* We consider three Multiobjective Evolutionary Algorithms: Nondominated Sorting Genetic Algorithm II (NSGAII) [9], Strength Pareto Evolutionary Algorithm 2 (SPEA2) [10], and Indicator-Based Evolutionary Algorithm (IBEA) [11]. They have been applied in the literature to engineering problems. It classifies the population in fronts. Each individual is assigned a rank corresponding to its nondominance level. This method ensures that the best solutions will remain at the next iteration. Elitism is therefore already incorporated without requiring an external procedure. NSGAII further reduces the complexity of the ordering procedure, based on nondominance, of its predecessor NSGA and allows the preservation of diversity by means of a technique called *crowding*. SPEA2 is a variant of SPEA. It assigns fitness by considering for each individual the class of individuals that dominate it and the class of those that are dominated by the individual. SPEA2 uses also a "closest neighbor" technique that values the density to improve the search. Finally, IBEA incorporates indexes of multiobjective quality, providing an alternative to Pareto dominance as a guide in the search.

## 4. Implementation and Design of Experiments

For our experiments we used four instances drawn from real cases: C_1 (15 × 20 problem with 157 operations), C_2 (20 × 20, 242 operations), C_3 (20 × 25, 412 operations), and C_4 (25 × 25, 597 operations). For each one we took into account the characteristics of the buffer, namely, no-wait and nonrestricted. Once the appropriate number of generations for the evolutionary phase and the production configuration for the microsimulations are defined, we run the experiments using PISA (A Platform and Programming Language Independent Interface for Search Algorithms) [25]. The parameters and characteristics of the experiments are shown in Table 2. For IBEA we chose the *additive epsilon* index. The other parameters keep their PISA predefined values. For each problem, the algorithm was run 30 times. From the class of solutions obtained, the dominated ones were eliminated. The running time of problems C_1, C_2, and C_3 was less than 30 minutes (Processor: 2.1 GHz AMD Turion X2 Ultra Dual-Core, Memory: 4 GB 800 MHz DDR2). C_4, instead, took in average 85,71 minutes.

## 5. Results

We provide in the next subsections analyses of the Pareto frontiers and a comparison of the three algorithms by means of the Hypervolume and R2 indexes.

*5.1. Pareto Frontiers.* The frontiers obtained in our experiments are shown in Figures 3 to 6. The horizontal axis represents objective $f_1$, the left vertical one, $f_2$, while the right vertical, $f_3$. Comparing $f_1$ and $f_2$, the fronts of the three algorithms look alike, although IBEA generated a better distributed front. NSGAII, instead, generated an incomplete frontier. With respect to $f_1$ versus $f_3$, we see that for C_1 with a buffer of 0 operations, algorithms SPEA2 and IBEA obtained better values than NSGAII (Figure 3). For C_1 with a buffer of 14 operations, IBEA got the best variance values (Figure 3). For C_2 the 0 operations buffer makes no difference (Figure 4). For C_2 with a buffer of 19 operations, IBEA yielded the best values in variance (Figure 4). On C_3 with a 0 operations buffer, NSGAII and IBEA got better values than SPEA2 (Figure 5), while for the 19 operations buffer, NSGAII and SPEA2 obtained better values than IBEA (Figure 5). Finally, on C_4, a 0 or 24 operations buffer made no difference (Figure 6).

*5.2. Comparison through Quality Indexes.* We compared the results according two indexes: Hypervolume [26] and R2 [27]. These are the usually recommended approaches to the evaluation of Pareto fronts. They provide slightly different advantages in the assessment of the frontiers. On one hand, Hypervolume seems fitter because it satisfies strong monotonicity while R2 only weak monotonicity. On the other, the former tends to be biased towards boundary solutions, while *R*2 is more uniform. Hypervolume requires a reference point to establish the area dominated by a given point, represented by the vector of its $f_1$, $f_2$, and $f_2$ values. Thus, a higher index indicates that the algorithm yields better solutions. *R*2 estimates the degree of closeness of the solution to the real front. Therefore, a low index indicates that the algorithm yields better solutions. Figures 7 and 8 show cases

Table 2: Parameters and characteristics of the experiments.

| | C_1<br>15 × 20, 157 op. | C_2<br>20 × 20, 242 op. | C_3<br>20 × 25, 412 op. | C_4<br>25 × 25, 597 op. |
|---|---|---|---|---|
| Inicialization | Random | Random | Random | Random |
| Representation | Binary | Binary | Binary | Binary |
| Number of genes | 20 | 20 | 25 | 25 |
| Size of gene | 22 | 62 | 62 | 84 |
| Size of chromosome | 820 | 1240 | 1550 | 2100 |
| Size of population | 50 | 50 | 100 | 100 |
| Generations | 1000 | 1000 | 1000 | 1000 |
| Crossover type | Uniform | Uniform | Uniform | Uniform |
| Probability of crossover | 0.85 | 0.85 | 0.85 | 0.85 |
| Mutation type | Alterate | Alterate | Alterate | Alterate |
| Mutated alleles | 82 | 124 | 155 | 210 |
| Probability of mutation | 0.05 | 0.05 | 0.05 | 0.05 |
| Number of objectives | 3 | 3 | 3 | 3 |
| Number of runs | 30 | 30 | 30 | 30 |
| Buffer limits | 0–14 | 0–19 | 0–19 | 0–24 |

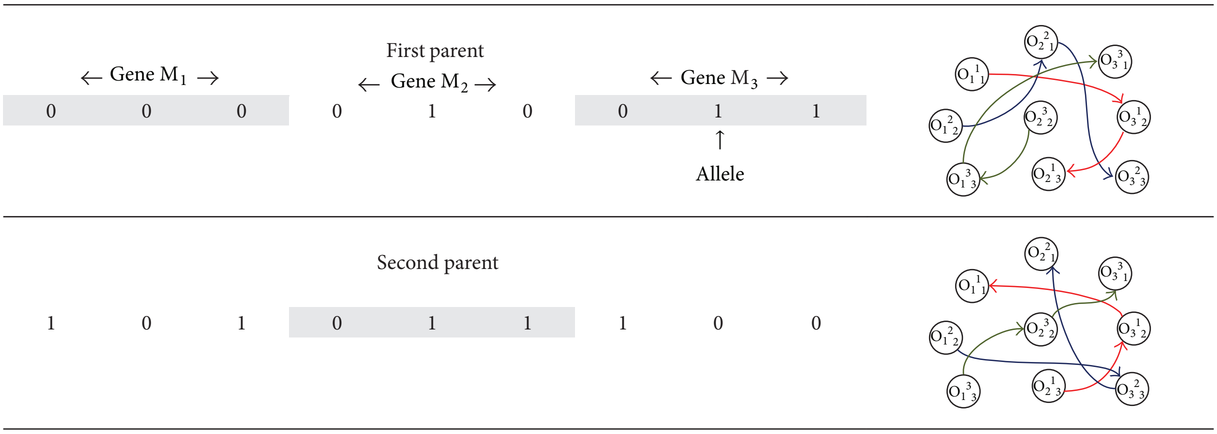


Figure 1: Chromosomes representing schedules.

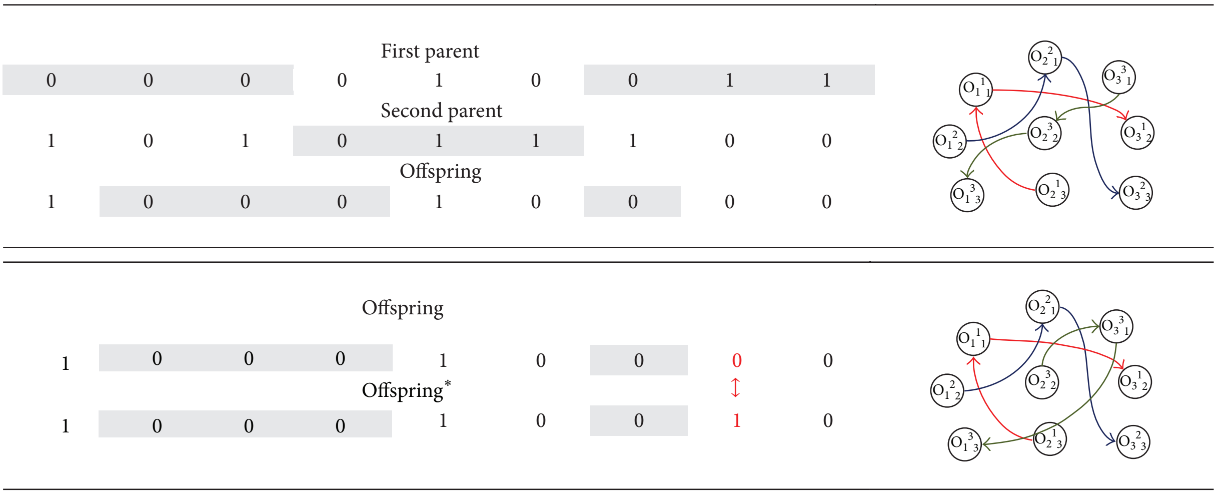


Figure 2: Crossover and mutation.

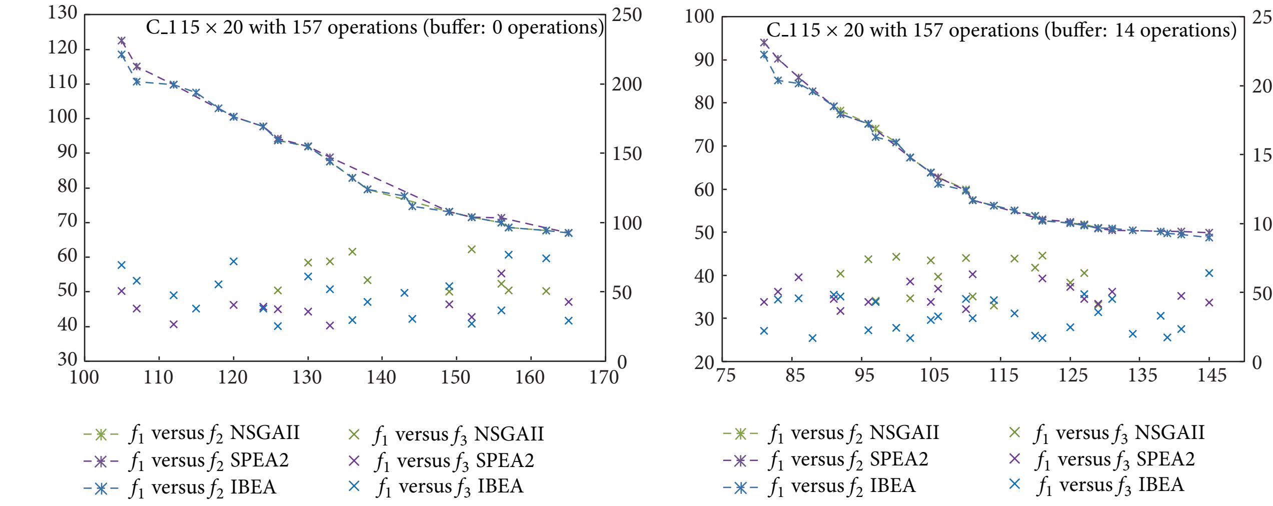


Figure 3: Pareto frontiers, $f_1$ versus $f_2$ and $f_1$ versus $f_3$, C_1.

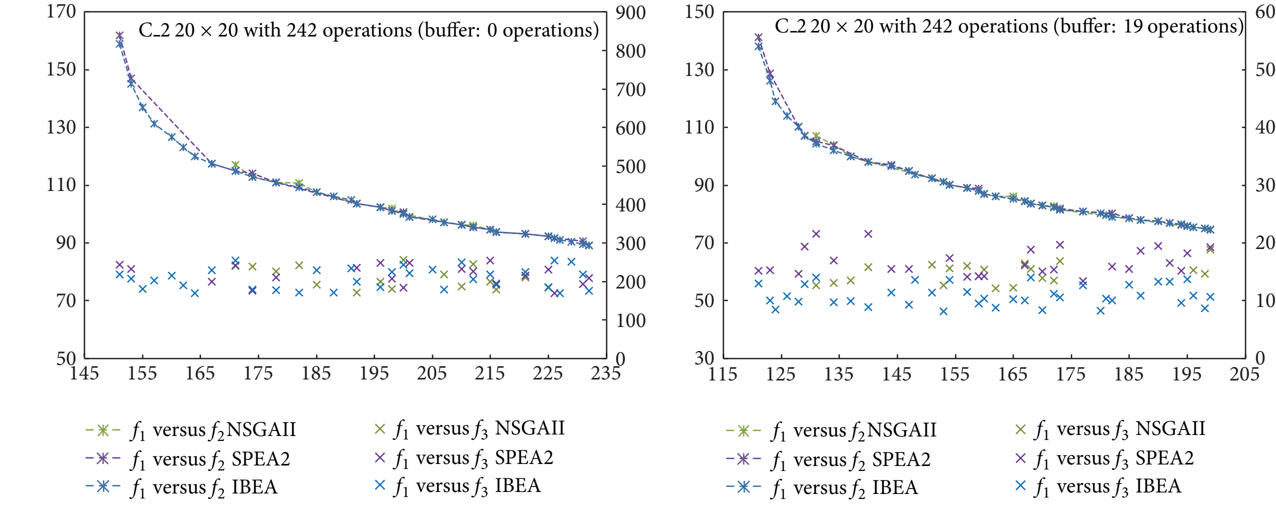


Figure 4: Pareto frontiers, $f_1$ versus $f_2$ and $f_1$ versus $f_3$, C_2.

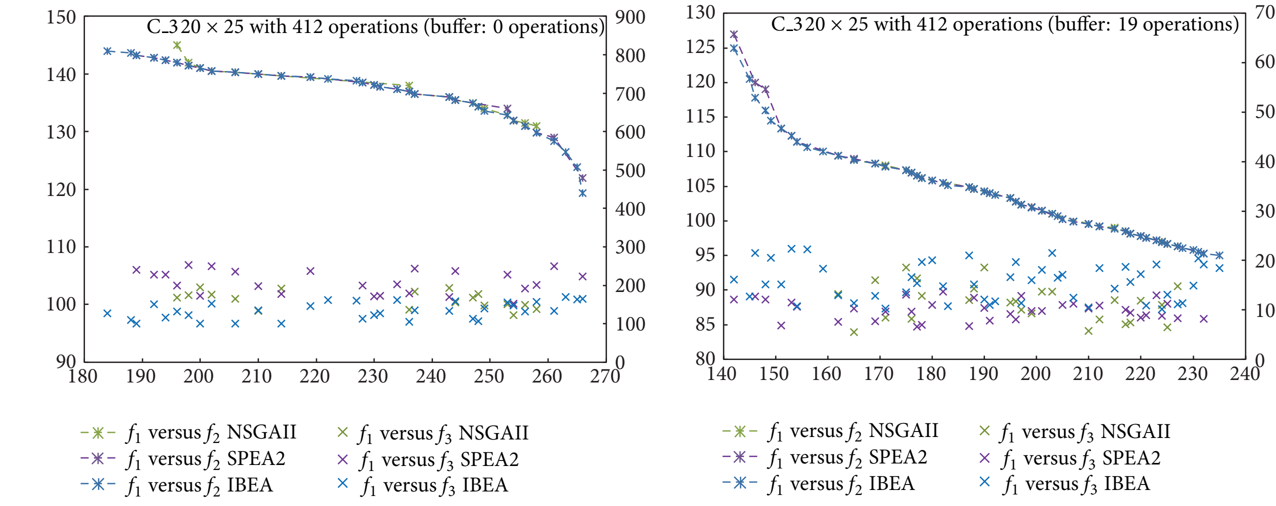


Figure 5: Pareto frontiers, $f_1$ versus $f_2$ and $f_1$ versus $f_3$, C_3.

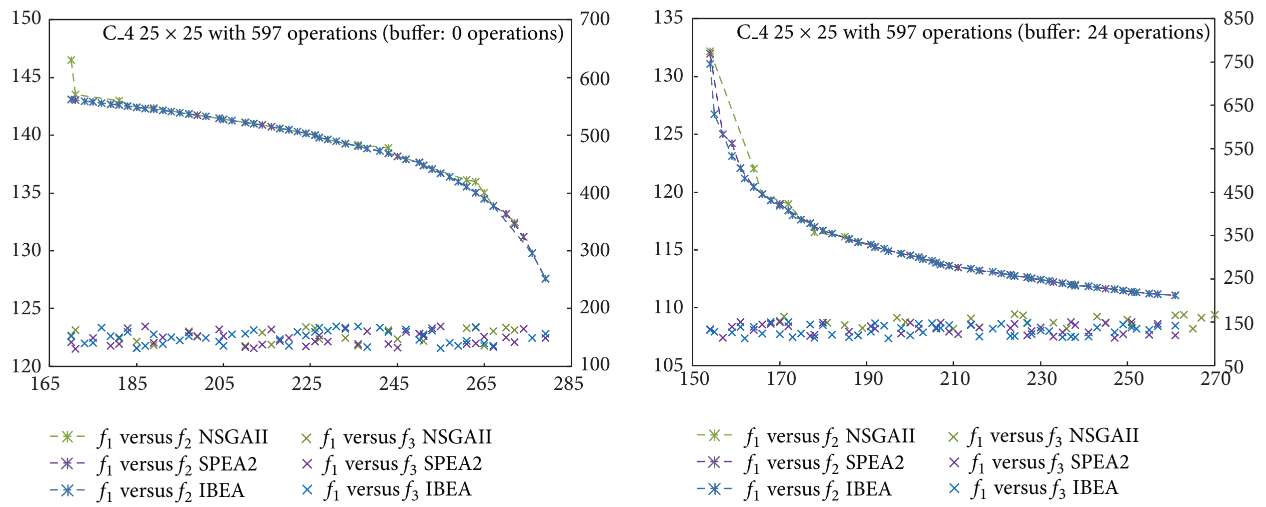


Figure 6: Pareto frontiers, $f_1$ versus $f_2$ and $f_1$ versus $f_3$, C_4.

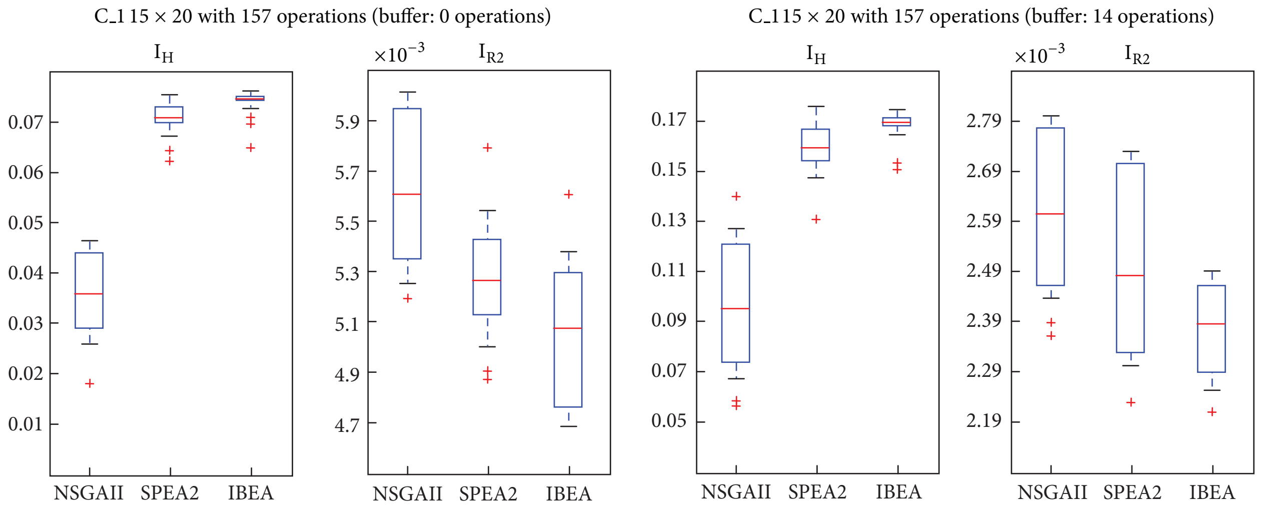


Figure 7: Boxplot/values for NSGAII, SPEA2, and IBEA, C_1.

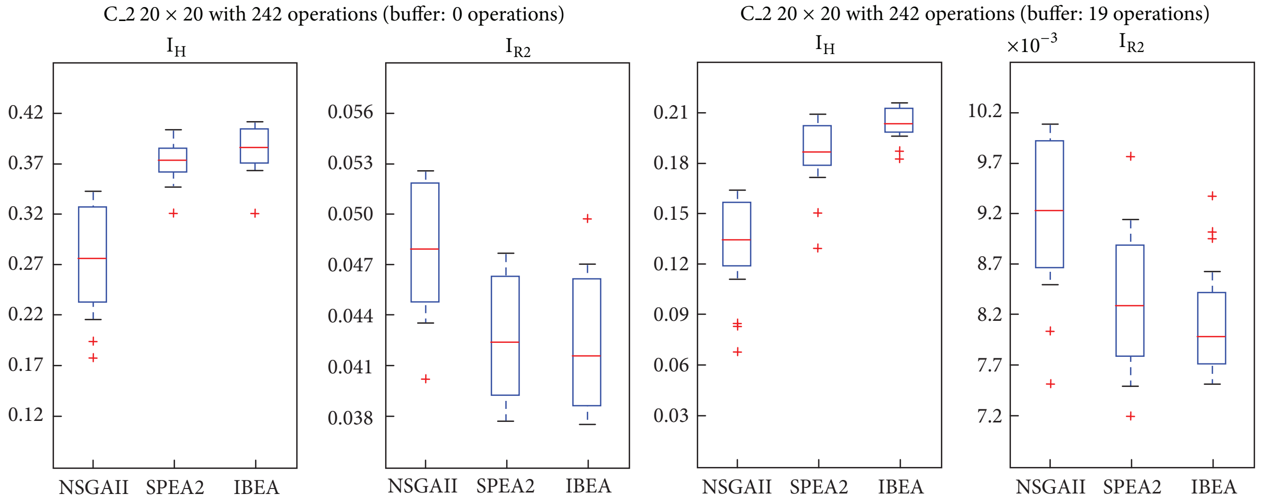


Figure 8: Boxplot/values for NSGAII, SPEA2, and IBEA, C_2.

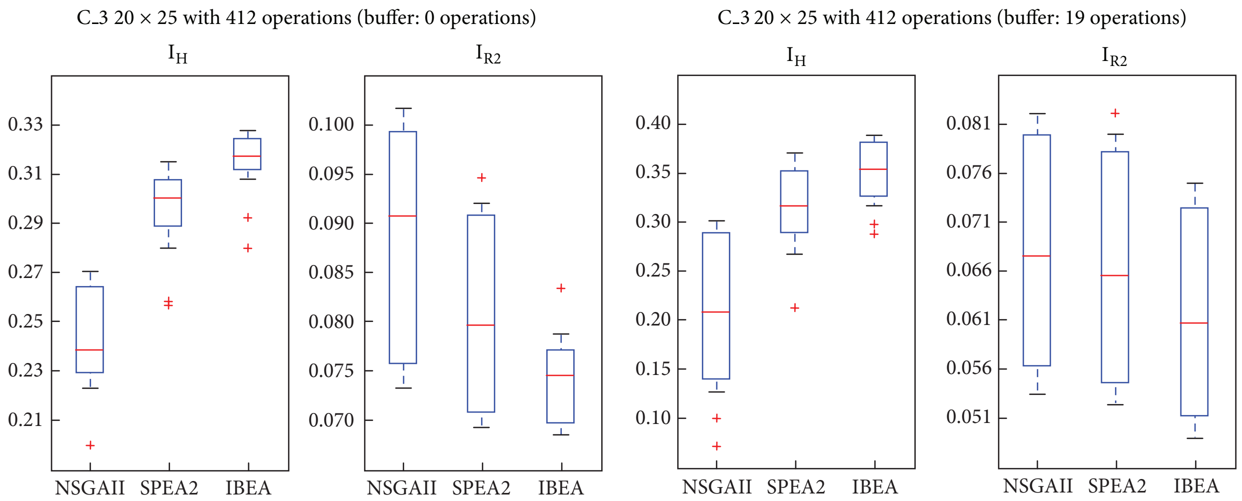


Figure 9: Boxplot/values for NSGAII, SPEA2, and IBEA, C_3.

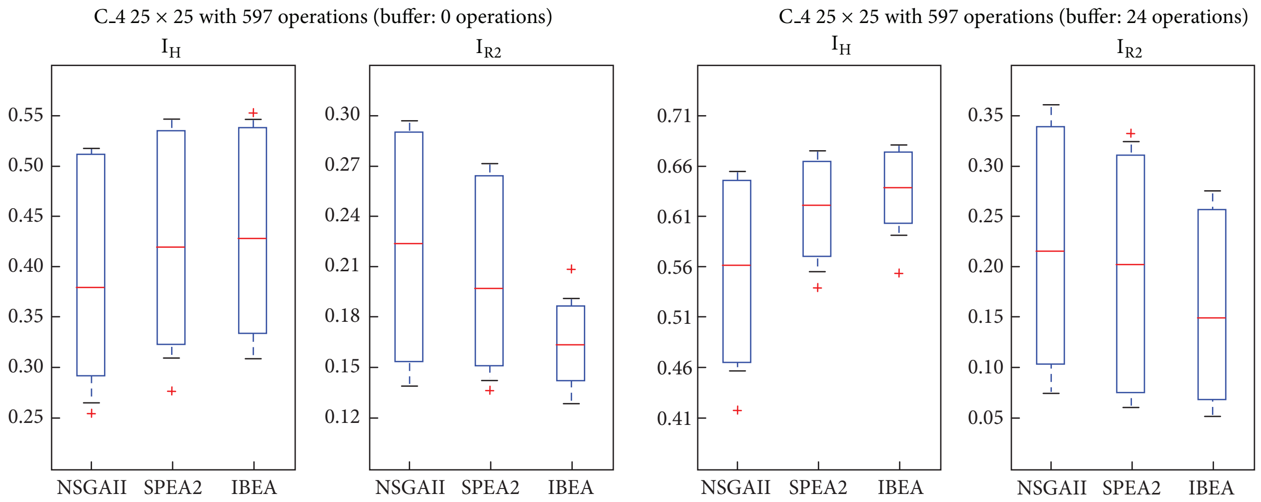


Figure 10: Boxplot/values for NSGAII, SPEA2, and IBEA, C_4.

Table 3: Approximate Pareto frontier and the contribution of each algorithm.

| Problem | NSGAII | SPEA2 | IBEA |
|---|---|---|---|
| C_1 (no-wait) | 50,00% | 60,00% | 100,00% |
| C_1 (non-restricted) | 59,26% | 66,67% | 100,00% |
| C_2 (no-wait) | 48,48% | 60,61% | 100,00% |
| C_2 (non-restricted) | 51,28% | 69,23% | 100,00% |
| C_3 (no-wait) | 52,94% | 73,53% | 100,00% |
| C_3 (non-restricted) | 50,98% | 74,51% | 100,00% |
| C_4 (no-wait) | 46,30% | 77,78% | 85,19% |
| C_4 (non-restricted) | 56,14% | 73,68% | 91,23% |

in which SPEA2 and IBEA are better according both indexes. Figure 9, instead, shows a case in which IBEA yields the better results, while Figure 10 presents a case in which there are no differences among the three algorithms. A possible explanation for NSGAII's general low degree of efficiency is that more than 2 objectives impair the crowding operator. Besides, it is well known that this algorithm is not efficient with binary representations.

To these casual observations we added a parametric statistical analysis, Fisher's test, with a significance level of 0.05. Problems C_1, C_2, and C_3 present significant differences in favor of SPEA2 and IBEA over NSGAII. Even if C_3 IBEA seems to perform better than SPEA2, the statistical analysis does not yield differences between these two algorithms. Finally, in C_4 there are no significant differences among the algorithms. Table 3 shows the contribution of each algorithm to the approximate Pareto, formed by taking the nondominated solutions.

## 6. Conclusions

This paper presents an analysis of the performance of three different Multiobjective Evolutionary Algorithms in experiments with Job-Shop Scheduling Problems. It required the specification of parameters appropriate for the problems at hand, involving constraints on machine availability and buffer capacity. An important share of the running time of the algorithms corresponded to microsimulations of the variance of makespan of solutions. The comparison leads to the selection of SPEA2 and IBEA, while the contribution to the approximate Pareto frontier makes IBEA the most efficient algorithm for the problems at hand. Future work involves the extension of this comparison on other production environment problems. It seems also worthwhile analyzing the implications of the variance of makespan in comparison with other objectives.

## Acknowledgments

The authors want to thank the Consejo Nacional de Investigaciones Científicas y Técnicas of Argentina (CONICET), the Engineering Department of the Universidad Nacional del Sur (PGI 24/J056), the Agencia Nacional de Promoción Científica y Tecnológica (PICT-2011-0396), and the Institute of Intelligent Systems and Numerical Applications in Engineering (IUSIANI) of the Universidad de Las Palmas de Gran Canaria (Spain), for their support.

## References

[1] R. Bihlmaier, A. Koberstein, and R. Obst, "Modeling and optimizing of strategic and tactical production planning in the automotive industry under uncertainty," *OR Spectrum*, vol. 31, no. 2, pp. 311–336, 2009.

[2] J. Chao-Hsien and H. Han-Chiang, "A hybrid genetic algorithm for no-wait job shop scheduling problems," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5800–5806, 2009.

[3] J. D. Ullman, "NP-complete scheduling problems," *Journal of Computer and System Sciences*, vol. 10, no. 3, pp. 384–393, 1975.

[4] L. Chinyao and Y. Yuling, "Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 314–322, 2009.

[5] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York, NY, USA, 1989.

[6] C. A. C. Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic, New York, NY, USA, 2002.

[7] B. J. Park, H. R. Choi, and H. S. A. Kim, "A hybrid genetic algorithm for the job shop scheduling problems," *Computers & Industrial Engineering*, vol. 45, no. 4, pp. 597–613, 2003.

[8] M. Frutos and F. Tohmé, "A multi-objective memetic algorithm for the job-shop scheduling problem," *Operational Research*, vol. 13, no. 2, pp. 233–250, 2012.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm for multi-objective optimization," in *Evolutionary Methods for Design, Optimisations and Control*, K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, Eds., pp. 19–26, 2002.

[11] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature—PPSN VIII*, X. Yao, E. K. Burke, and J. Lozano, Eds., vol. 3242 of *Lecture Notes in Computer Science*, pp. 832–842, Springer, 2004.

[12] V. A. Armentano and C. R. Scrich, "Tabú search for minimizing total tardiness in a job shop," *International Journal of Production Economics*, vol. 63, no. 2, pp. 131–140, 2000.

[13] S. Binato, W. J. Hery, D. M. Loewenstern, and M. G. C. Resende, "A GRASP for job shop scheduling, essays and surveys," in *Meta-Heuristics*, pp. 59–80, Kluwer Academic, Boston, Mass, USA, 2000.

[14] J. Heinonen and F. Pettersson, "Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 989–998, 2007.

[15] I. Kacem, S. Hammadi, and P. Borne, "Correction to 'approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems'," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 32, no. 2, 172 pages, 2002.

[16] P. Fattahi, M. Saidi, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 18, no. 3, pp. 331–342, 2007.

[17] C. F. Tsai and F. C. Lin, "A new hybrid heuristic technique for solving job-shop scheduling problem," in *Proceedings of the 2nd IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems*, pp. 53–58, 2003.

[18] C. G. Wu, X. L. Xing, H. P. Lee, C. G. Zhou, and Y. C. Liang, "Genetic algorithm application on the job shop scheduling problem," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 4, pp. 2102–2106, 2004.

[19] N. B. Ho and J. C. Tay, "Evolving dispatching rules for solving the flexible job-shop problem," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 5, pp. 2848–2855, 2005.

[20] N. B. Ho, J. C. Tay, and E. M. Lai, "An effective architecture for learning and evolving flexible job-shop schedules," *European Journal of Operational Research*, vol. 179, no. 2, pp. 316–333, 2007.

[21] H. Zhang and M. Gen, "Multistage-based genetic algorithm for flexible job-shop scheduling problem," *Complexity International*, vol. 11, pp. 223–232, 2005.

[22] F. Pezzella, G. Morganti, and G. Ciaschetti, "A genetic algorithm for the flexible job-shop scheduling problem," *Computers & Operations Research*, vol. 35, no. 10, pp. 3202–3212, 2008.

[23] T. C. Chiang and H. J. Lin, "A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling," *International Journal of Production Economics*, vol. 141, no. 1, pp. 87–98, 2013.

[24] J. R. Evans and D. L. Olson, *Introduction to Simulation and Risk Analysis*, Editorial Prentice Hall, NJ, USA, 1998.

[25] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "PISA—a platform and programming language independent interface for search algorithms," in *Evolutionary Multi-Criterion Optimization*, vol. 2632, pp. 494–508, 2003.

[26] E. Zitzler and L. Thiele, "Multi-objective optimization using evolutionary algorithms, a comparative case study," in *Parallel*

*Problem Solving From Nature—PPSN V*, A. E. Eiben, Ed., pp. 292–301, Springer, Amsterdam, The Netherlands, 1998.

[27] M. P. Hansen and A. Jaszkiewicz, "Evaluating the quality of approximations to the non-dominated set," Tech. Rep. IMM-REP-1998-7, Institute of Mathematical Modelling Technical University of Denmark, 1998.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 125893, 10 pages
http://dx.doi.org/10.1155/2013/125893

*Research Article*

# A Hybrid Genetic-Simulated Annealing Algorithm for the Location-Inventory-Routing Problem Considering Returns under E-Supply Chain Environment

**Yanhui Li,[1] Hao Guo,[1] Lin Wang,[2] and Jing Fu[1,3]**

[1] *School of Information Management, Central China Normal University, Wuhan 430079, China*
[2] *School of Management, Huazhong University of Science and Technology, Wuhan 430074, China*
[3] *Department of Industrial and System Engineering, State University of New York at Buffalo, Buffalo, NY 14228, USA*

Correspondence should be addressed to Lin Wang; wanglin982@gmail.com

Received 4 October 2013; Accepted 18 November 2013

Academic Editors: T. Chen, Q. Cheng, and J. Yang

Facility location, inventory control, and vehicle routes scheduling are critical and highly related problems in the design of logistics system for e-business. Meanwhile, the return ratio in Internet sales was significantly higher than in the traditional business. Many of returned merchandise have no quality defects, which can reenter sales channels just after a simple repackaging process. Focusing on the existing problem in e-commerce logistics system, we formulate a location-inventory-routing problem model with no quality defects returns. To solve this NP-hard problem, an effective hybrid genetic simulated annealing algorithm (HGSAA) is proposed. Results of numerical examples show that HGSAA outperforms GA on computing time, optimal solution, and computing stability. The proposed model is very useful to help managers make the right decisions under e-supply chain environment.

## 1. Introduction

The increasing progress of information and prevalence of internet in the 21st century have forced the e-commerce to develop in world-wide rage. In 2012, B2C e-commerce sales grew 21.1% to top $1 trillion for the first time in history of the whole world [1]. Comparing with traditional commerce, customers are liable to return goods under e-commerce environment. Note that many customer returns online accounts for 35% of original orders [2, 3]. Therefore, logistics systems as an important support system in e-commerce need to be adjusted and improved. To adapt to the reality of e-commerce market environment, reverse logistics network and highly integrated logistics process should be the necessities.

Facility location, inventory control, and vehicle routing decisions are critical problems in the design of logistics system. There is much previous work on these three areas. Furthermore, the related work on location and vehicle routing was extended into the field of computer communication and networks [4, 5]. In fact, there is a mutually dependent relationship among these problems in logistics system. Comprehensive optimizing and logistics activities management should be based on this relationship [6]. According to this idea, besides location allocation problem and vehicle routing problem, two-two integration such as location-routing problem (LRP), inventory-routing problem (IRP), and location-inventory problem (LIP) and three integration problem (location-inventory-routing problem, LIRP) start to be researched.

Many papers about the LIP, LRP, and IRP are studied deeply and have made some abundant achievements. However, research on the integration of location-inventory-routing problem is limited. Some researchers strongly appeal to carry out research on LIRP [7, 8]. Liu and Lee [9] firstly proposed the LIRP; they built a model for single merchandise, multi-DPs LRP taking inventory control decisions into consideration and proposed a two-stage heuristic algorithm. In order to avoid being trapped in local optima, Liu and Lin [10] proposed a global optimum heuristic based on the algorithm in the above papers to solve the LIRP. Max Shen and Qi

[11] established a nonlinear integer programming model to minimize the total cost that includes location costs, inventory costs, and transportation costs and proposed a Lagrangian relaxation based algorithm to solve the model. Ahmadi Javid and Azad [12] presented an LIRP model in a stochastic supply chain system and established a heuristic method based on a hybridization of tabu search and simulated annealing to solve the LIRP model. Ahmadi-Javid and Seddighi [13] considered the LIRP of a multisource distribution logistics network. A mixed-integer programming formulation was presented and a three-phase heuristic was developed to solve the problem.

Previously, reverse logistics mainly researched independent activities about LIRP; Fleischmann et al. [14] and Jayaraman et al. [15] are interested in determining the location of recycling center with capacity constraints. In recent years, some researches on reverse logistics concerned the integrated system. Lieckens and Vandaele [16] applied a queuing mode in reverse logistics network to solve the facility location problem while considering the impact of inventory costs. Sahyouni et al. [17] developed three generic facility location models that account for both forward and reverse logistics network; Easwaran and Üster [18] proposed a mixed-integer linear programming model to optimize the total cost that consists of location, processing, and transportation costs of the multimerchandise closed-loop supply chains; Srivastava [19] established a reverse logistics network optimization model to optimize the location-distribution problem and capacity decisions, and he pointed out that integrated optimization of processing, storage, transportation, and recycling merchandises is one of the directions of future research.

Previous researches on the reverse logistics system optimization mainly focus on the minimization of the total cost in forward logistics network. To our best knowledge, researches on manufacturing/remanufacturing system by taking customer returns and concept of green logistics recycling into account in reverse logistics are very limited. Since the fact that customers may dissatisfy with merchandise and return it, the cost of processing returns, the cost of inventory and delivery, ordering time, and quantity are changed.

The aim of this study is to develop a practical LIRP model with considering returns under e-supply chain environment and provide a new hybrid heuristic algorithm. To our best knowledge, this work is the first step to introduce returns into the LIRP under e-supply chain environment, which makes it become more practical. We also provide an effective algorithm named hybrid genetic simulated annealing algorithm (HGSAA) to solve this model. Results of numerical examples show that HGSAA outperforms genetic algorithm (GA) on computing time, optimal solution, and computing stability.

The remainder of this paper is organized as follows. In Section 2, a nonlinear integrated programming model based on forward and reverse logistics networks about LIRP is proposed under e-supply chain environment. Section 3 designs the heuristic algorithm named HGSAA. Section 4 contains the results of different experiments and corresponding analysis. Section 5 proposes conclusions and future research directions.

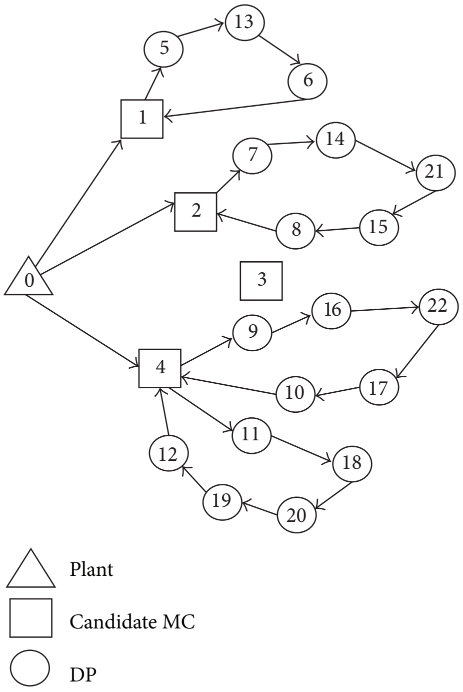


Figure 1: The specific network graph.

## 2. Mathematical Model

*2.1. Problem Description.* In e-supply chain network, returned merchandises in sales generally have a high integrity, which makes them usually do not need to be repaired and can reenter the sales channels after a simple repackaging process [20]. Therefore, distribution centers and recycling centers can be merged into merchandise centers (MCs). MC is responsible for distributing normal goods to the demand points (DPs) of downstream, meanwhile the returned goods are collected to MCs. After repackaging treatment at MCs, returned goods become resalable normal goods.

Based on the above, the supply chain in this study consists of one plant, multiple MCs, and multiple DPs, which is a three-phases (production base, merchandise centers, and demand points) e-commerce logistics system. Considering the return policy in e-commerce, we optimize system construction, operation of the facility location, inventory control, and coordinate arrangements of vehicle routing.

The operations of product order and returns are as follows. Previously, the finished productions are transferred from the plant to the MCs. Then the merchandises are delivered to DPs, which in turn collect returned merchandises. Returned merchandises are processed and repackaged in MCs and then sold as normal goods. The operations mode is shown in Figure 1.

The objective of this problem is to determine the quantity, locations, order times, and order size of MCs and arrange the routes that vehicles visiting the DPs in the integrated logistics network. The final target is to minimize the total cost and improve the efficiency of logistics operations. The involved decisions are as follows: (1) location decisions, the optimal number of MCs and their locations; (2) inventory decisions, the optimal order times and order size on a route; (3) routing arrangement, the vehicles deliver merchandises and collect returned merchandises in the order.

### *2.2. Assumptions*

(1) There is a single type of merchandise.

(2) The total demand on each route is less than or equal to the vehicle capacity.

(3) The vehicle type is homogeneous.

(4) Each route is served by one vehicle.

(5) Each route begins and ends at the same MC.

(6) The capacity of MCs is infinite.

(7) The forward distribution and reverse collection service could be met at the same time.

(8) The daily demand and return of each DP are known.

(9) The returned merchandises are without quality defect.

(10) Returned merchandises are processed and repackaged at MCs.

*2.3. Model Formulation and Analysis.* The cost of MC$_r$ consists of the following components.

(i) The annual cost of the dispatching vehicle at MC$_r$ is given by $N_r e_r$.

(ii) The annual cost of placing an order at MC$_r$ is given by $N_r f_r$.

(iii) As for the returned merchandise without quality defects, it can be sold again as a normal goods being repackaged; therefore, transportation volume from plant to MC$_r$ shall deduct the returns $\sum_{i\in\mathbf{S}} q_i$. Thus, the annual transportation cost from plant to MC$_r$ is given by $\lambda \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} b_r(d_i - q_i)Y_{ir}^{v}$.

(iv) As the same reason as the third point above, the annual inventory holding costs at MC$_r$ should consider the returns, too. So, the annual inventory holding cost at MC$_r$ is given by $(\lambda \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} h(d_i + q_i)Y_{ir}^{v})/2N_r$. Since there is $\sum_{i\in\mathbf{S}}(d_i - q_i)$ in component (iii), and here has $\sum_{i\in\mathbf{S}}(d_i + q_i)$, adding this two parts together, that is $\sum_{i\in\mathbf{S}}(d_i - q_i) + \sum_{i\in\mathbf{S}}(d_i + q_i) = \sum_{i\in\mathbf{S}} d_i$. It means the goods flow is equal to demand.

(v) The annual handling cost at MC$_r$ is given by $\lambda \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} c_r d_i Y_{ir}^{v}$.

(vi) The annual repackaging cost of returned merchandises at MC$_r$ is given by $\lambda \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} p q_i Y_{ir}^{v}$.

(vii) The annual total distribution costs from every MC to each DP is given by $N_r \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}^+} \sum_{j\in\mathbf{S}^+} \sum_{v\in\mathbf{V}} l s_{ij} X_{ijr}^{v}$.

(viii) The construction cost of MC$_r$ is given by $\sum_{i\in\mathbf{R}} a_r Z_r$.

The objective is to minimize the total cost of the system; we formulate the model as follows:

$$\begin{aligned}\min Z = {} & \sum_{r\in\mathbf{R}} (e_r + f_r) N_r + \lambda \sum_{r\in\mathbf{R}} \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} b_r (d_i - q_i) Y_{ir}^{v} \\ & + \frac{\lambda \sum_{r\in\mathbf{R}} \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} h (d_i + q_i) Y_{ir}^{v}}{2N_r} \\ & + \lambda \sum_{r\in\mathbf{R}} \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} c_r d_i Y_{ir}^{v} + \lambda \sum_{r\in\mathbf{R}} \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} p q_i Y_{ir}^{v} \\ & + N_r \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}^+} \sum_{j\in\mathbf{S}^+} \sum_{v\in\mathbf{V}} l s_{ij} X_{ijr}^{v} + \sum_{r\in\mathbf{R}} a_r Z_r\end{aligned} \tag{1}$$

s.t.

$$Z_r \geq 1, \quad r \in \mathbf{R}; \tag{2}$$

$$\sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}^+} X_{ijr}^{v} = 1, \quad j \in \mathbf{S}; \tag{3}$$

$$\sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} Y_{ir}^{v} = 1, \quad i \in \mathbf{S}; \tag{4}$$

$$\sum_{j\in\mathbf{S}^+} X_{kjr}^{v} - \sum_{i\in\mathbf{S}^+} X_{ikr}^{v} = 0, \quad k \in \mathbf{S}^+, \quad v \in \mathbf{V}, \quad r \in \mathbf{R}; \tag{5}$$

$$\sum_{i\in\mathbf{S}} \sum_{r\in\mathbf{R}} d_i Y_{ir}^{v} \leq g, \quad v \in \mathbf{V}; \tag{6}$$

$$Y_{ir}^{v} - Z_r \leq 0, \quad r \in \mathbf{R}, \quad i \in \mathbf{S}, \quad v \in \mathbf{V}; \tag{7}$$

$$\sum_{j\in\mathbf{S}^+} X_{ijr}^{v} + \sum_{j\in\mathbf{S}^+} X_{rjr}^{v} - Y_{ir}^{v} \leq 1, \quad i \in \mathbf{S}, \quad v \in \mathbf{V}, \quad r \in \mathbf{R}; \tag{8}$$

$$Z_r = \{0, 1\}, \quad r \in \mathbf{R}; \tag{9}$$

$$Y_{ir}^{v} = \{0, 1\}, \quad r \in \mathbf{R}, \quad i \in \mathbf{S}, \quad v \in \mathbf{V}; \tag{10}$$

$$X_{ijr}^{v} = \{0, 1\}, \quad i \in \mathbf{S}, \quad j \in \mathbf{S}^+, \quad r \in \mathbf{R}, \quad v \in \mathbf{V}, \tag{11}$$

where the objective function (1) minimizes the system's total cost; (2) ensures at least one MC is established; (3) ensures each DP is served by the only one vehicle which belongs to a certain MC; (4) ensures that each route has only one vehicle; (5) ensures the continuity of delivery routes; (6) ensures vehicle cannot be overloaded; (7) ensures that only the selected MC can carry out distribution services; (8) ensures as long as a route passing through a DP, the corresponding MC would also be on this route; (9)–(11) ensure the integrality of decision variables.

## 3. Solution Approach

In this section, we first give the formula for solving optimal order times $N_r$ and the optimal order size $Q_r^{v}$. Since calculating $N_r$ and $Q_r^{v}$ still relies on the decision variables $X_{ijr}^{v}$, $Y_{ir}^{v}$, and $Z_r$, so we present a heuristic algorithm to get the optimized $X_{ijr}^{v}$, $Y_{ir}^{v}$, and $Z_r$.

*3.1. Finding the Optimal Order Times.* In the models (1)–(11), the decision variable $N_r$ only has appeared in the objective function. Also, the objective function is convex for $N_r > 0$. Consequently, we can obtain the optimal value of $N_r$ by taking the derivative of the objective function with respect to $N_r$ as

$$N_r = \sqrt{\frac{\sum_{v\in \mathbf{V}}\sum_{i\in \mathbf{S}} \lambda h\left(d_i + q_i\right) Y_{ri}^{v}}{2\left(e_r + f_r + \sum_{i\in \mathbf{S}^{+}}\sum_{j\in \mathbf{S}^{+}}\sum_{v\in \mathbf{V}} ls_{ij} X_{ijr}^{v}\right)}}. \qquad (12)$$

Then, the optimal order size can be given by

$$\begin{aligned} Q_r &= \frac{\sum_{v\in \mathbf{V}}\sum_{i\in \mathbf{S}} \lambda d_i Y_{ri}^{v}}{N_r} \\ &= \left( \sum_{v\in \mathbf{V}}\sum_{i\in \mathbf{S}} \lambda d_i Y_{ri}^{v} \right. \\ &\qquad \left. \times \left( \sqrt{\frac{\sum_{v\in \mathbf{V}}\sum_{i\in \mathbf{S}} \lambda h\left(d_i + q_i\right) Y_{ri}^{v}}{2\left(e_r + f_r + \sum_{i\in \mathbf{S}^{+}}\sum_{j\in \mathbf{S}^{+}}\sum_{v\in \mathbf{V}} ls_{ij} X_{ijr}^{v}\right)}} \right)^{-1} \right). \end{aligned} \qquad (13)$$

*3.2. Hybrid Genetic Simulated Annealing Algorithm (HGSAA).* The LIRP contains the VRP. As we know, the VRP is an NP-hard problem. This makes LIRP more complicated. It is generally believed that there is no complete, accurate, and not too slow analytic algorithm to solve NP-hand problems. Noting bioinspired computation is widely used for solving optimization problems, we designed a hybrid algorithm based on GA and simulated annealing (SA) to solve the proposed model.

Traditional GA has strong global search ability in solving such problems, but also has defects such as premature and weak local search ability. On the other hand, SA has strong local search ability and no premature problem. Therefore, the combination of GA and SA can overcome the defects of each of the two methods, bring into play their respective advantages, and improve the solving efficiency. This algorithm is named hybrid genetic simulated annealing algorithm (HGSAA).

*3.2.1. Relevant Operations of GA*

*(1) Encoding.* In a genetic algorithm, a population of candidate solutions (called individuals) to an optimization problem is evolved toward better solutions. Each individual with a set of properties, such as its chromosomes or genotype, can be mutated and altered. Traditionally, solutions are represented in binary as strings of 0 s and 1 s.

This study adopts the natural number coding method; using unrepeatable $R + S$ natural numbers constitute a sequence, which represents an individual. While $1, 2, \ldots, R$ indicate candidate MCs and $R + 1, \ldots, R + S$ indicate DPs, the encoding can describe a candidate solution of the above optimization problem. For example, in Figure 1, the code of the individual corresponding solutions is {1, 5, 13, 6, 2, 7, 14, 21, 15, 8, 3, 4, 9, 16, 22, 17, 10, 11, 18, 20, 19, 12}.

*(2) Fitness Function.* A fitness function is a particular type of objective function that is used to measure the quality of the represented solution. In this study, the fitness function is defined as

$$f_k = \frac{1}{Z}. \qquad (14)$$

*(3) Selection.* During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Wheel selection operator [21] (also known as proportional selection operator) is used. Suppose the population size is $N$, the fitness value of the individual $k$ is $f_k$, a number $\xi \in [0, 1]$ is generated randomly. If $(\sum_{k=1}^{i-1} f_k / \sum_{k=1}^{N} f_k) < \xi \le (\sum_{k=1}^{i} f_k / \sum_{k=1}^{N} f_k)$ then individual $i$ is selected to be replicated.

Because this method has great randomness in the selection of individuals, the simulated annealing algorithm with faster local convergence is added to the GA to increase the convergence speed in selection operation.

*(4) Crossover.* Crossover is a process of taking more than one parent individuals and producing a child individual from them. Crossover is used to vary the programming of a chromosome or chromosomes from one generation to the next. Partially matched crossover (PMX) [22] is used in this paper.

*Step 1.* Select two parent individuals randomly from the population;

*Step 2.* generate two random cut points to represent the mapped segments;

*Step 3.* exchange the segments of the two parent individuals to produce two new individuals;

*Step 4.* determine the mapping relations between two segments;

*Step 5.* legalize two new individuals with mapping relationship through repair strategy.

*(5) Mutation.* Mutation is used to maintain genetic diversity from one generation of a population of individuals to the next. The purpose of mutation in GAs is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of individuals from becoming too similar to each other, thus slowing or even stopping evolution. A simple and efficient mutation operation, that is, swap mutation [23], is used. The details are as follows.

*Step 1.* Select one parent individual randomly from the population;

*Step 2.* generate two random numbers to represent the mutation points;

*Step 3.* swap the positions of these two mutation points to produce a new individual.

Compared with other mutations, studies show that convergence rate of this method has a greater advantage in population control. It can effectively prevent premature convergence of GA and avoid the occurrence of local optimal solution.

*3.2.2. Relevant Operations of SA*

*(1) The Annealing Process to Accept the New Individual.* In order to prevent the population into local optimization, the Metropolis acceptance criteria in SA are applied into the GA in this paper. We reserved the best parent individual in a population named old, and then selected the best offspring individual in another population named new; old and new go into the next generation population through competition. Let $\Delta_f = f_{\text{new}} - f_{\text{old}}$, if $\Delta_f < 0$, then the individual *new* is received, and hold it to the next generation; otherwise, the individual new is received with the probability $p = \exp(-\Delta_f/t) >$ random digit, where $t$ is annealing temperature.

*(2) Temperature Amended Criterion.* One of the key steps in the process of SA is to determine the update function of temperature; the function is used to continuously reduce the temperature value, when its temperature is reduced to approximately zero, the final solution is considered as the global optimal solution. The update function is $t_{k+1} = \alpha t_k$, $k \geq 0$, $0 < \alpha < 1$; the nearer $\alpha$ is to 1, the slower the temperature decreases.

*3.2.3. Termination.* Commonly, the algorithm terminates when either a maximum number of generations have been produced, or a satisfactory fitness level has been reached for the population. In this paper, the termination condition is that the fitness has reached a plateau such that successive $M$ iterations no longer produce better results.

*3.2.4. Algorithm Flow*

*Step 1.* Set the initial parameters: coordinates of the DPs and the candidate MCs, demands and returns of the DPs, the maximum capacity of the vehicle $g$, the population size $N$, evolution terminate generation $M$, crossover probability $p_c$, mutation probability $p_m$, temperature of the cooling coefficient $\alpha$, the initial annealing temperature $T_0$, and so on.

*Step 2.* Calculate the fitness value of an individual. If the parent optimal solution and offspring optimal solution are equal during continuous $M$ generations, the algorithm stops and outputs the current optimal solution; otherwise, go to the next step.

*Step 3.* Perform individual selection, crossover, and mutation operations, generate new population, and calculate the fitness value.

*Step 4.* If $f_i < f_j$ $(j > i)$, accept the new individual; otherwise, accept the new individual with the probability $p = \exp(-\Delta_f/t)$;

*Step 5.* Update the annealing temperature and return to Step 2.

The pseudocodes of HGSAA are shown in Pseudocode 1.

## 4. Computational Experiments and Results Analysis

*4.1. An Example.* An example is used to illustrate the proposed heuristic method. The data of Gaskell 67-29×5 come from the LRP database at University of Aveiro [24]. Gaskell 67 is the name of this instance; 29×5 means there are 29 DPs and 5 candidate MCs. The coordinate of all nodes and the demands of DPs are given by the database. To facilitate the calculation, the daily demands of DPs are set as 1/25 of corresponding demands in the LRP database. The other data are as follows: the inventory holding cost per unit of merchandise per year $h = 5$; the vehicle capacity $g = 500$; the delivering cost per unit distance $l = 1$; the handling cost per unit product at $\text{MC}_r$ $c_r = 4$; fixed cost of dispatching vehicles per time at $\text{MC}_r$ $e_r = 18$; the conversion constant $\lambda = 300$; repackaging cost of unit returned merchandise $p = 3$; $q_i$, $b_r$, $f_r$ are uniformly generated from $U[1,5]$, $U[6,10]$, $U[16,20]$.

According to the experience of literatures [25, 26], the related parameters of the HGSAA are set as follows: the population size $N = 20$; crossover probability $p_c = 0.8$; mutation probability $p_m = 0.001$; evolution terminate generation $M = 300$; initial temperature $T_0 = 100$; temperature cooling coefficient $\alpha = 0.9$.

Based on Matlab 6.5 platform, we programmed the HGSAA and then run it 30 times on a computer (CPU: Intel Core 2 Duo P8400 @ 2.26 GHz 2.27 GHz; RAM: 3 GB DDR; OS: Windows Vista). One of the minimum values of objective function is 1404100; the individual is encoded as {5, 3, 34, 23, 21, 33, 11, 32, 4, 12, 28, 19, 29, 14, 25, 13, 22, 27, 7, 15, 26, 8, 6, 18, 30, 2, 1, 10, 20, 24, 31, 9, 16, 17}, and $N_r = \{56, 0, 46, 94, 0\}$, $Q_r^v = \{333, 231, 444, 300, 409, 368\}$. Figure 2 shows the trends of optimal objective function values along with the evolution generations. Table 1 shows the solution.

For comparison, GA is programmed by Matlab 6.5 as well, and the instance Gaskell 67 was run 30 times on the same computer. The optimal objective function values of these two algorithms are shown in Table 2, the CPU time for calculation is shown in Table 3.

Figure 3 shows the trends of optimal objective function value along with the evolution generations by GA.

The fluctuation curves of optimal objective function values in 30 times are shown in Figures 4 and 5, respectively.

```
Procedure: HGSAA for LIRP
Input: coordinates of nodes, demands and returns of DPs, MC parameters, vehicle capacity,
       HGSAA parameters
Output: the best solution (include routes, MCs locations, order times and order size)
Begin
Take pop
  for  $i = 1$ to pop_size
      Calculate individual fitness value $f_i$
    end
    $f_{\text{best}} = \max\{f_i\}$
  $n = 1$
  $k = 1$
while $n \leq 300$
        select operator
          if  random $\leq p_c$  then
                crossover operator
          end
          if  random $\leq p_m$  then
                mutation operator
          end
        newpop
                for  $i = 1$ to newpop_size
                      Calculate individual fitness value $f_i$
                end
        $f_{\text{newbest}} = \max\{f_i\}$
        $f_{\text{worst}} = \min\{f_i\}$
                      if  $f_{\text{newbest}} > f_{\text{best}}$  then
                         $n = 1$
                         $f_{\text{best}} = f_{\text{newbest}}$
                         $pop(\text{best},:) = newpop(\text{newbest},:)$
                      elseif  $f_{\text{newbest}} == f_{\text{best}}$
                            $n = n + 1$
                      else
                            $n = 1$
                         $\Delta = \frac{1}{f_{\text{newbest}}} - \frac{1}{f_{\text{best}}}$
                         $z = e^{(-\Delta/t)}$
                              if  $z >$ random  then
                                 $f_{\text{best}} = f_{\text{newbest}}$
                                 $pop(\text{best},:) = newpop(\text{newbest},:)$
                              end
                      $T = \alpha T$
                         end
        $newpop(\text{worst},:) = pop(\text{best},:)$
        $pop = newpop$
        $k = k + 1$
  end
  output  the best solution
end
```

Pseudocode 1: Pseudocode of the proposed HGSAA.

Table 1: Solution of Gaskell 67-29×5.

| MC | Routing number | Routing | Order times | Order quantity |
|---|---|---|---|---|
| 1 | 1 | $i_{10}$-$i_{20}$-$i_{24}$-$i_{31}$-$i_9$ | 56 | 333 |
| | 2 | $i_{16}$-$i_{17}$ | 56 | 231 |
| 3 | 3 | $i_{34}$-$i_{23}$-$i_{21}$-$i_{33}$-$i_{11}$-$i_{32}$ | 46 | 444 |
| | 4 | $i_{12}$-$i_{28}$-$i_{19}$-$i_{29}$-$i_{14}$-$i_{25}$-$i_{13}$-$i_{22}$-$i_{27}$ | 94 | 300 |
| 4 | 5 | $i_7$-$i_{15}$ | 94 | 409 |
| | 6 | $i_{26}$-$i_8$-$i_6$-$i_{18}$-$i_{30}$ | 94 | 368 |

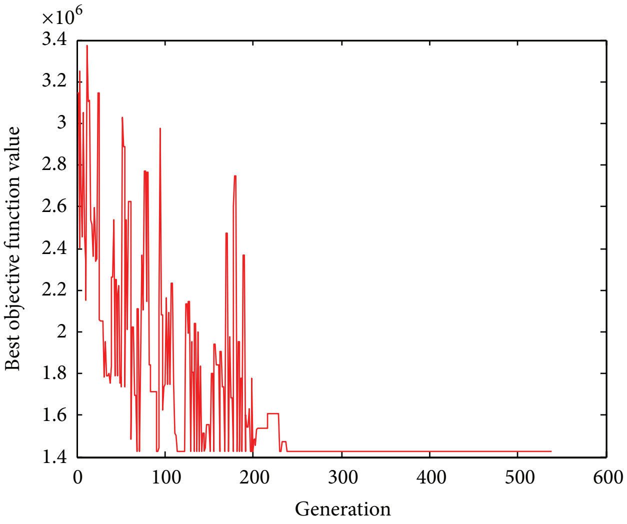


Figure 2: Trends of optimal objective function value by HGSAA.

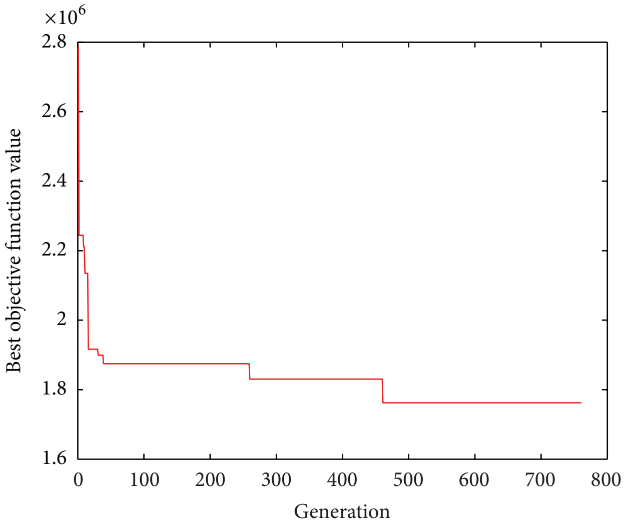


Figure 3: Trends of optimal objective function value by GA.

Figures 2 and 3 show that HGSAA can converge to the optimal solution more quickly than GA. Moreover, HGSAA has better stability than GA, which can easily be found from Tables 2 and 3 and Figures 4 and 5.

*4.2. Extended Experiments.* In this section, a series of experiment is given to show that HGSAA is more efficient and stable than GA. Similarly as Section 4.1, all the experiments in this section come from LRP database of the University of Aveiro [24]. In order to ensure the demands of DPs are not more than the vehicle capacity, we need to enumerate some instances. In this study, the daily demands are set as 1/15 of corresponding demands of Gaskell 67-22×5.

Results of numerical example in Section 4.1 show that the related parameters of HGSAA in Gaskell 67-22×5 are reasonable. Thus, we employ these parameters in the remainder of this section. Each instance was calculated 30 times by HGSAA and GA, respectively; the results are shown in Tables 4 and 5.

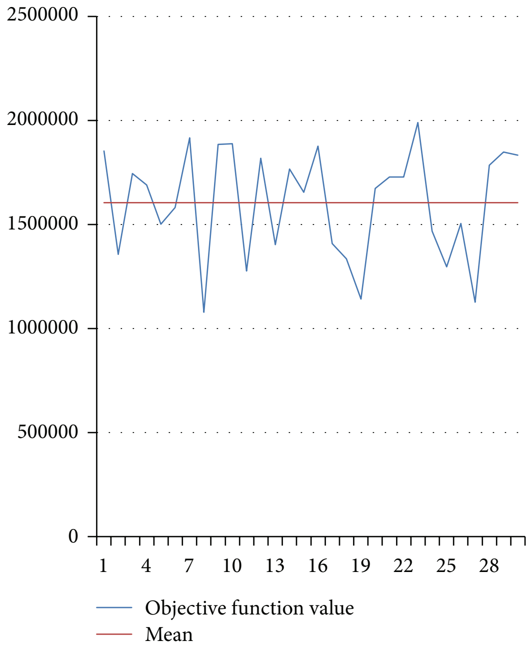


Figure 4: The fluctuation curve of optimal objective function value by HGSAA.

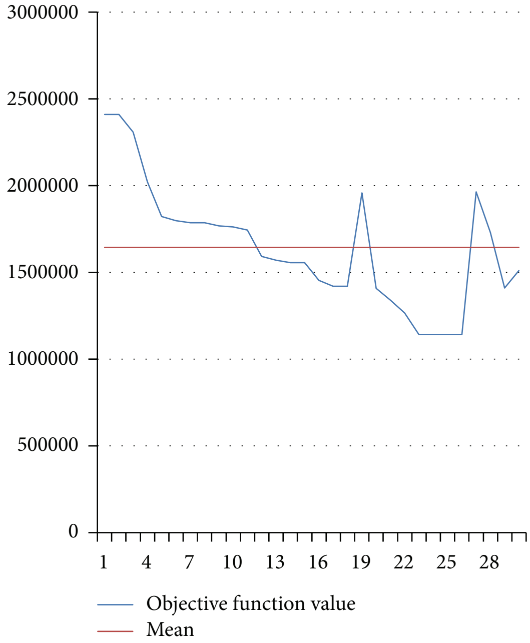


Figure 5: The fluctuation curve of optimal objective function value by GA.

Table 4 shows that HGSAA can obtain better objective function value than GA. Table 5 shows that HGSAA takes less time to achieve the optimal solution than GA. Results of Tables 4 and 5 show that HGSAA is more stable than GA.

Table 2: Statistical results of optimal objective function value of two algorithms.

| | Maximum | Minimum | Mean | Standard deviation | Coefficient of variation |
|---|---|---|---|---|---|
| HGSAA | 1989900.00 | 1078700.00 | 1605543.33 | 262462.64 | 0.16 |
| GA | 2410900.00 | 1142000.00 | 1644678.00 | 353993.10 | 0.22 |

Table 3: Statistics results of CPU time for calculation of two algorithms.

| | Maximum | Minimum | Mean | Standard deviation | Coefficient of variation |
|---|---|---|---|---|---|
| HGSAA | 5.93 | 3.35 | 4.68 | 0.80 | 0.17 |
| GA | 7.35 | 3.63 | 4.93 | 1.18 | 0.24 |

## 5. Conclusion and Future Research

Under the e-commerce environment, customers have a higher return rate. At the same time, the returned goods have generally no quality defect and with great integrity. Just after a simple repackaging process, the returned goods can reenter the sales channels, which put forward high requirements to the logistics system that support the operation of e-commerce. This study handles the above interesting problem and provides an effective heuristic. The main contributions are as follows.

(1) In reality, the cost of processing returned merchandises is produced considering the customers are not satisfied with products and maybe return them. We firstly design a LIRP model to minimize the total cost produced by both forward and reverse logistics networks. It is very useful to help managers make the right decision under e-supply chain environment.

(2) An integration LIRP model with returns is an NP-hard problem and very hard to be solved by analytical method. So, a heuristic algorithm named HGSAA is designed by integrating GA with SA.

(3) Results of experimental data show that HGSAA outperforms GA on computing time, optimal solution, and computing stability. HGSAA is a good candidate to solve the proposed LIRP model effectively.

However, some extensions should be considered in further work. Considering the variety of the types of products and service vehicles, the multiple products and multiple vehicles type model should be established. In reality, decision makers are always in front of imprecise and vague operational conditions [27]. Uncertainties have been tackled in a lot of ways and fuzzy set theory has a long history for handle imprecise values [28]. Considering the fuzzy demand of customs or related fuzzy costs, more practical LIRP model should be developed. Moreover, differential evolution algorithms (DEs) have turned out to be one of the best evolutionary algorithms in a variety of fields [29]. In the future, we may use an improved DE to find better solutions for the LIRPs. The integration research and practice of the management of e-commerce logistics system can be constantly improved.

## Notations

*Sets*

$\mathbf{R}$: Set of candidate MC

$\mathbf{S}$: Set of DPs

$\mathbf{S}^+$: Set of candidate MCs and DPs; that is, $\mathbf{S}^+ = \mathbf{S} \cup \mathbf{R}$

$\mathbf{V}$: Set of routes are available for the routing from the MCs to DPs.

*Constants*

$a_r$: Fixed (annual) administrative and operational cost of $\mathrm{MC}_r$

$b_r$: The unit of product transportation cost from plant to $\mathrm{MC}_r$

$c_r$: The handling costs per unit product at $\mathrm{MC}_r$

$d_i$: Mean (daily) demand at $\mathrm{DP}_i$

$e_r$: Fixed cost of dispatching vehicles per time at $\mathrm{MC}_r$

$f_r$: Fixed cost of placing an order at $\mathrm{MC}_r$

$g$: The vehicle capacity

$h$: The inventory holding cost per unit of merchandise per year

$l$: The delivering cost per unit distance

$p$: Repackaging cost of unit returned merchandise at MC

$q_i$: The quantity of merchandises returned by $\mathrm{DP}_i$ per day

$s_{ij}$: The distance from depot $i$ to depot $j$

$\lambda$: Working days per year.

*Decision Variables*

$N_r$: Optimal order times at $\mathrm{MC}_r$

$Q_r^v$: Optimal order size on routing $v$ for $\mathrm{MC}_r$

$X_{ijr}^v$: 1, if depot $j$ is from depot $i$ served by a $\mathrm{MC}_r$ on routing $v$, and 0 otherwise

$Y_{ir}^v$: 1, if $\mathrm{DP}_i$ is assigned to $\mathrm{MC}_r$ on routing $v$, and 0 otherwise

$Z_r$: 1, if candidate $\mathrm{MC}_r$ is selected as a MC location, and 0 otherwise.

Table 4: Optimal objective function values of two algorithms.

| Instance name | Algorithm | Maximum | Minimum | Mean | Standard deviation | Coefficient of variation |
|---|---|---|---|---|---|---|
| Perl 183-12×2 | HGSAA | 740420 | 157810 | 502684 | 176218.7 | 0.350555 |
| | GA | 1322000 | 219800 | 660282 | 234416.0 | 0.355024 |
| Gaskell 67-22×5 | HGSAA | 1718000 | 1071600 | 1432110 | 213118.8 | 0.148815 |
| | GA | 3146500 | 1506200 | 2170660 | 553894.4 | 0.255173 |
| Gaskell 67-36×5 | HGSAA | 3635500 | 1877000 | 2879510 | 561628.4 | 0.195043 |
| | GA | 3836900 | 2219400 | 2886460 | 582830.7 | 0.201919 |
| Perl 183-55×15 | HGSAA | 4120700 | 3606800 | 3985110 | 152566.1 | 0.038284 |
| | GA | 4307000 | 3755100 | 4090590 | 185978.4 | 0.045465 |
| Christofides 69-75×10 | HGSAA | 5562400 | 4290900 | 4826600 | 394532.4 | 0.081741 |
| | GA | 6359300 | 4859800 | 5418878 | 525869.4 | 0.097044 |
| Perl 83-85×7 | HGSAA | 6529200 | 5050500 | 5693300 | 531113.4 | 0.093287 |
| | GA | 7057200 | 5545800 | 6283210 | 428004.8 | 0.068119 |
| Christofides 69-100×10 | HGSAA | 5978900 | 5074600 | 5592260 | 332403.7 | 0.05944 |
| | GA | 6177500 | 5211900 | 5792190 | 354180.4 | 0.061148 |

Table 5: CPU time (seconds) for calculation of two algorithms.

| Instance name | Algorithm | Maximum | Minimum | Mean | Standard deviation | Coefficient of variation |
|---|---|---|---|---|---|---|
| Perl 183-12×2 | HGSAA | 0.8424 | 0.4992 | 0.65156 | 0.085708 | 0.131543 |
| | GA | 0.9672 | 0.5304 | 0.6696 | 0.102803 | 0.153529 |
| Gaskell 67-22×5 | HGSAA | 1.716 | 1.092 | 1.37436 | 0.169531 | 0.123353 |
| | GA | 2.1216 | 1.17 | 1.55688 | 0.272075 | 0.174756 |
| Gaskell 67-36×5 | HGSAA | 3.7596 | 2.2776 | 3.2058 | 0.501429 | 0.156413 |
| | GA | 4.8984 | 2.5584 | 3.81108 | 0.760696 | 0.199601 |
| Perl 183-55×15 | HGSAA | 12.0121 | 8.5957 | 9.91249 | 0.999262 | 0.100808 |
| | GA | 12.6517 | 9.8125 | 10.91837 | 0.846487 | 0.077529 |
| Christofides 69-75×10 | HGSAA | 13.3225 | 10.1713 | 11.74674 | 1.162717 | 0.098982 |
| | GA | 13.7437 | 10.2805 | 12.30382 | 1.370948 | 0.111425 |
| Perl 83-85×7 | HGSAA | 21.6373 | 17.0509 | 18.94194 | 1.305246 | 0.068908 |
| | GA | 24.3518 | 17.8621 | 21.04736 | 2.279007 | 0.10828 |
| Christofides 69-100×10 | HGSAA | 33.2906 | 30.1859 | 31.36279 | 0.93519 | 0.029818 |
| | GA | 33.5558 | 30.6386 | 32.21258 | 0.962075 | 0.029866 |

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (nos. 70871050; 71171093; 71371080; 71131004; and 70801030).

## References

[1] "Ecommerce sales topped $1 trillion for the first time in 2012," http://www.emarketer.com/Article/Ecommerce-Sales-Topped-1-Trillion-First-Time-2012/1009649.

[2] H. Meyer, "Many happy returns," *Journal of Business Strategy*, vol. 20, no. 4, pp. 27–31, 1999.

[3] C. R. Gentry, "Reducing the cost of returns," *Chain Store Age*, vol. 75, no. 10, pp. 124–126, 1999.

[4] J. Yang and Z. Fei, "Statistical filtering based broadcast protocol for vehicular networks," in *Proceedings of the 20th International Conference on Computer Communications and Networks (ICCCN '11)*, pp. 1–6, Hawaii, USA, July 2011.

[5] J. Yang and Z. Fei, "HDAR: hole detection and adaptive geographic routing for ad hoc networks," in *Proceedings of the 19th International Conference on Computer Communications and Networks (ICCCN '10)*, pp. 1–6, Zurich, Switzerland, August 2010.

[6] C. Watson-Gandy and P. Dohrn, "Depot location with van salesmen: a practical approach," *Omega*, vol. 1, no. 3, pp. 321–329, 1973.

[7] G. Nagy and S. Salhi, "Location-routing: issues, models and methods," *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.

[8] N. H. Moin and S. Salhi, "Inventory routing problems: a logistical overview," *Journal of the Operational Research Society*, vol. 58, no. 9, pp. 1185–1194, 2007.

[9] S. C. Liu and S. B. Lee, "A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration," *International Journal of Advanced Manufacturing Technology*, vol. 22, no. 11-12, pp. 941–950, 2003.

[10] S. C. Liu and C. C. Lin, "A heuristic method for the combined location routing and inventory problem," *International Journal*

*of Advanced Manufacturing Technology*, vol. 26, no. 4, pp. 372–381, 2005.

[11] Z.-J. Max Shen and L. Qi, "Incorporating inventory and routing costs in strategic location models," *European Journal of Operational Research*, vol. 179, no. 2, pp. 372–389, 2007.

[12] A. Ahmadi Javid and N. Azad, "Incorporating location, routing and inventory decisions in supply chain network design," *Transportation Research E*, vol. 46, no. 5, pp. 582–597, 2010.

[13] A. Ahmadi-Javid and A. H. Seddighi, "A location-routing-inventory model for designing multisource distribution networks," *Engineering Optimization*, vol. 43, no. 10, pp. 1–19, 2012.

[14] M. Fleischmann, P. Beullens, J. M. Bloemhof-Ruwaard, and L. N. van Wassenhove, "The impact of product recovery on logistics network design," *Production and Operations Management*, vol. 10, no. 2, pp. 156–173, 2001.

[15] V. Jayaraman, R. A. Patterson, and E. Rolland, "The design of reverse distribution networks: models and solution procedures," *European Journal of Operational Research*, vol. 150, no. 1, pp. 128–149, 2003.

[16] K. Lieckens and N. Vandaele, "Reverse logistics network design with stochastic lead times," *Computers and Operations Research*, vol. 34, no. 2, pp. 395–416, 2007.

[17] K. Sahyouni, R. C. Savaskan, and M. S. Daskin, "A facility location model for bidirectional flows," *Transportation Science*, vol. 41, no. 4, pp. 484–499, 2007.

[18] G. Easwaran and H. Üster, "Tabu search and Benders decomposition approaches for a capacitated closed-loop supply chain network design problem," *Transportation Science*, vol. 43, no. 3, pp. 301–320, 2009.

[19] S. K. Srivastava, "Network design for reverse logistics," *Omega*, vol. 36, no. 4, pp. 535–548, 2008.

[20] D. Vlachos and R. Dekker, "Return handling options and order quantities for single period products," *European Journal of Operational Research*, vol. 151, no. 1, pp. 38–52, 2003.

[21] S. Jakobs, "On genetic algorithms for the packing of polygons," *European Journal of Operational Research*, vol. 88, no. 1, pp. 165–181, 1996.

[22] M. Gen, R. Cheng, and L. Lin, *Network Models and Optimization: Multi-Objective Genetic Algorithm Approach*, Springer, Heidelberg, Germany, 2008.

[23] T. W. Leung, C. H. Yung, and M. D. Troutt, "Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem," *Computers and Industrial Engineering*, vol. 40, no. 3, pp. 201–214, 2001.

[24] "Location-Routing Problems (LRP)," http://sweet.ua.pt/~iscf143/_private/SergioBarretoHomePage.htm.

[25] H. H. Orkcu, "Subset selection in multiple linear regression models: a hybrid of genetic and simulated annealing algorithms," *Applied Mathematics and Computation*, vol. 129, no. 23, pp. 11018–11028, 2013.

[26] Y. Xu, R. Qu, and R. F. Li, "A simulated annealing based genetic local search algorithm for multi-objective multicast routing problems," *Annals of Operations Research*, vol. 206, no. 1, pp. 527–555, 2013.

[27] L. Wang, Q. L. Fu, C. G. Lee, and Y. R. Zeng, "Model and algorithm of fuzzy joint replenishment problem under credibility measure on fuzzy goal," *Knowledge-Based Systems*, vol. 39, pp. 57–66, 2013.

[28] L. Wang, C. X. Dun, C. G. Lee, Q. L. Fu, and Y. R. Zeng, "Model and algorithm for fuzzy joint replenishment and delivery scheduling without explicit membership function," *International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9–12, pp. 1907–1920, 2013.

[29] L. Wang, H. Qu, Y. H. Li, and J. He, "Modeling and optimization of stochastic joint replenishment and delivery scheduling problem with uncertain costs," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 657465, 12 pages, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 728414, 12 pages
http://dx.doi.org/10.1155/2013/728414

*Research Article*

# Applying Probability Theory for the Quality Assessment of a Wildfire Spread Prediction Framework Based on Genetic Algorithms

**Andrés Cencerrado, Ana Cortés, and Tomàs Margalef**

*Computer Architecture and Operating Systems Department, Autonomous University of Barcelona, Bellaterra, 08193 Barcelona, Spain*

Correspondence should be addressed to Andrés Cencerrado; andres.cencerrado@uab.es

Received 1 October 2013; Accepted 18 November 2013

Academic Editors: T. Chen, Q. Cheng, and J. Yang

This work presents a framework for assessing how the existing constraints at the time of attending an ongoing forest fire affect simulation results, both in terms of quality (accuracy) obtained and the time needed to make a decision. In the wildfire spread simulation and prediction area, it is essential to properly exploit the computational power offered by new computing advances. For this purpose, we rely on a two-stage prediction process to enhance the quality of *traditional* predictions, taking advantage of parallel computing. This strategy is based on an adjustment stage which is carried out by a well-known evolutionary technique: Genetic Algorithms. The core of this framework is evaluated according to the probability theory principles. Thus, a strong statistical study is presented and oriented towards the characterization of such an adjustment technique in order to help the operation managers deal with the two aspects previously mentioned: time and quality. The experimental work in this paper is based on a region in Spain which is one of the most prone to forest fires: *El Cap de Creus*.

## 1. Introduction

As stated in [1], the potential for a natural hazard to become a disaster mainly depends on a society's capacity to address the underlying risk factors, to reduce the vulnerability of a community, and to be ready to respond in case of emergency. In the last years, the scientific community has provided many advances in order to deal with the issue of the early remote sensing [2–5], which represents a great advantage. However, most of these phenomena present an additional important problem regarding the uncertainty of the variables that describe the scenario where they take place. Forest fires are a clear example of this problem, making them a very complex system to model and simulate.

Over the last decades, different physical models have been developed and implemented into forest fire spread simulators. The Rothermel model [6] can be considered the most used and representative among them. The research studies regarding fire behavior are very valuable in order to develop and optimize computational methods so as to develop simulation and prediction tools, such as FireLib [7], FireStation [8], or FARSITE [9]. Furthermore, given the complexity of these implemented models as well as their implicit computational demands, there are different works that focused on the optimization of their results either by coupling different models [10] or by exploiting the increasing computational capabilities [11, 12].

Fire spread simulators usually require input parameters that in some cases are uniform on the whole terrain and constant on time, but some others vary from one point of the terrain to another one or present a temporal evolution. Furthermore, it is very difficult to gather accurate and reliable values of certain parameters at the right places where the catastrophe is taking place, because the hazard itself often distorts the measurements. So, in many cases the unique alternative consists of working with interpolated, outdated, or even absolutely unknown values. Obviously, this fact adds the problem of dealing with high levels of uncertainty in the input

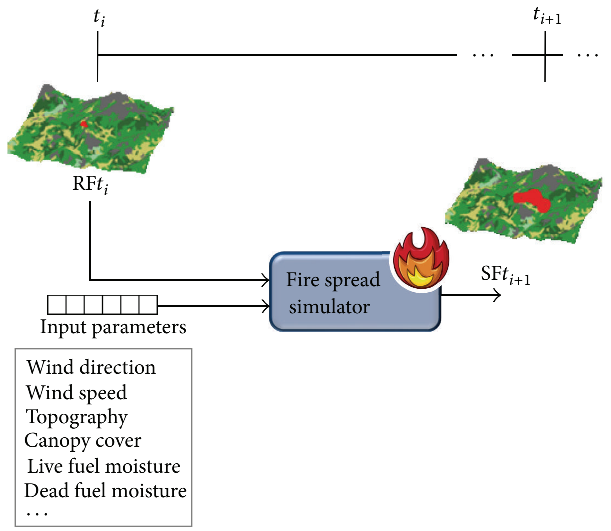


Figure 1: Classical prediction schema. RF and SF stand for *real fire* and *simulated fire*, respectively.

parameters, which results in a lack of accuracy and quality on the provided predictions.

To overcome the input data uncertainty, a two-stage prediction methodology was developed [13]. This methodology divides the prediction process into two different stages: adjustment and prediction. During the first stage, a calibration method is applied using data about the observed fire propagation to obtain the input data set which best describes the actual evolution of the catastrophe. Afterwards, the prediction stage starts a new simulation using the values obtained in the calibration stage. Subsequently, the differences between the classical prediction schema and the two-stage prediction method are detailed.

*1.1. Classical Prediction.* The traditional way of predicting forest fire behavior takes the initial state of the fire front as input as well as the input parameters given for that time instant. These values are entered into any existing fire simulator, which then returns the prediction for the state of the fire front at a later time instant. This is summarized in Figure 1, where $t_i$ stands for a certain time instant $i$, RF$t_i$ stands for the real fire spread at a certain time instant $i$, and SF$t_i$ stands for the simulated spread at a certain time instant $i$.

The forecasted fire front tends to differ from the real fire line to a greater or lesser extent. As the prediction error accumulates gradually as the prediction time advances, deviations between real phenomenon behavior and forecasted fire spread become even more significant. One reason for this incidence is that the classic calculation of the simulated fire is based upon one single set of input parameters afflicted with many inadequacies. To improve parameter quality and enable real-time estimation and calibration of model input parameters in each time step during an ongoing prediction, a two-stage prediction scheme was proposed in [13].

*1.2. Two-Stage Prediction Method.* Fire spread simulators need certain input data which define the characteristics of the environment where the fire is taking place in order to evaluate its future propagation. This data usually consists of the current fire front, terrain topography, vegetation type, and meteorological data such as humidity, wind direction, and wind speed. Some of this data could be retrieved in advance and with notable accuracy, for example, the topography of the area and the predominant vegetation types.

However, there is some data that turns out to be very difficult to be reliably obtained. For instance, getting an accurate fire perimeter is very complicated because of the difficulties involved in getting real-time images. Live and dead fuel moistures are examples of data which cannot be retrieved with reliability at the moment of the emergency. Another kind of data sensitive to imprecisions is that of meteorological data, which is often distorted by the fire itself. However, this circumstance is not only related to forest fires but also happens in any system with a dynamic state evolution over time, for example, floods [14], thunderstorms [15, 16], and so forth. These uncertainties, added to the fact that these inputs are set up only at the very beginning of the simulation process, become an important drawback because, as the simulation time goes on, variables previously initialized could change dramatically, misleading simulation results. In order to overcome these restrictions, we need a system capable of properly estimating the values of the input parameters needed by the underlying simulator so that the results we obtain correspond to reality.

Introducing a previous adjustment step, the set of input parameters is optimized before every prediction step. Figure 2 schematizes this process. In this figure, the ε process consists of the comparison between the actual and the simulated fire spreads. As it is explained in Section 3, for this calculation we rely on the *symmetric difference* between sets of burned cells.

Thus, the proposed solution comes from reversing the problem: how to find a parameter configuration such that, given this configuration as input, the fire simulator would produce predictions that match the actual fire behavior.

Having detected the simulator input that best describes current environmental conditions, the same set of parameters could also be used to best describe the immediate future, assuming that meteorological conditions remain constant during the next prediction interval. Then, the prediction becomes the result of a series of automatically adjusted input configurations.

The process of input parameters adjustment can be seen as a global optimization process, where we search for an instance in an N-dimensional space (N equals the number of input parameters to be adjusted) which minimizes the difference between the simulated fire spread produced and the real observed spread. There is a great variety of works which rely on the so-called Global Optimization Algorithms [17] to solve different problems of this kind [18–21]. In the specific case of forest fire spread, several adjustment techniques have been tested to calibrate the input parameter set, from which stands out the use of bioinspired evolutionary computation, specifically Genetic Algorithms (GAs), which

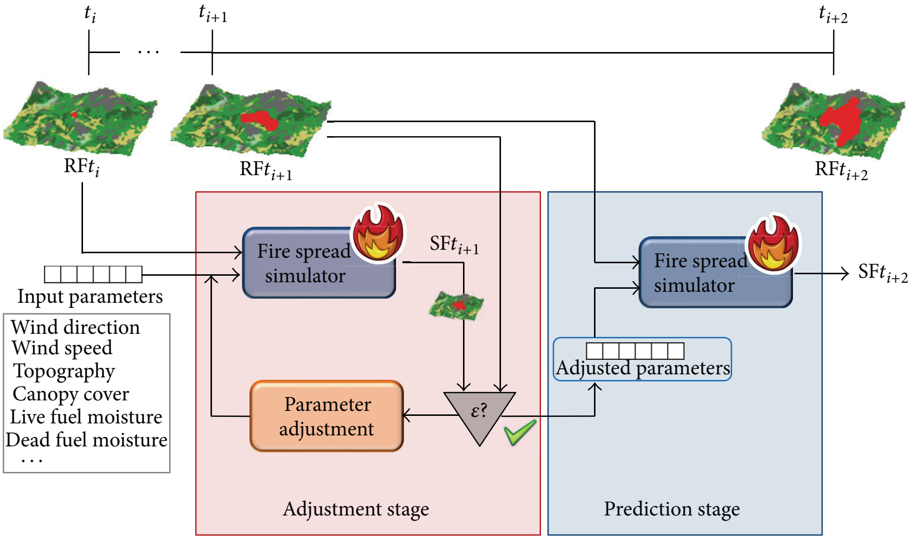


Figure 2: Two-stage prediction method.

has provided outstanding results as a parallel adjustment technique within the two-stage prediction framework [22].

In order to be useful, any evolution prediction of an ongoing hazard must be delivered as fast as possible in order not to be outdated. When relying on the two-stage prediction method, accuracy also depends on the amount of computational resources we have access to performing different simulations, since the use of parallel computation favors the evolutionary methods used in the adjustment stage. As it will be studied in Section 4, the fact that the adjustment strategies for parameter optimization may be carried out in a parallel way allows us to take advantage of the simultaneous execution of different simulations.

The rest of this paper is organized as follows. In the next section, we explain the proposed methodology to assess in advance the quality of the predictions. In Section 3, the details concerning the characterization of the GAs as adjustment strategy, based on the principles of probability theory, are given. Section 4 presents a statistical study based on the area of *El Cap de Creus* (northeast of Spain) in order to validate our propose methods. Finally, the main conclusions are given in Section 5.

## 2. Methodology for Time Response and Quality Assessment for Forest Fire Spread Prediction

From the point of view of attending an ongoing emergency, it is necessary to be able to assess, in advance, the amount of computational resources needed to deal with. This is due to the existing strict deadlines for giving a response. Moreover, it is also necessary to accurately assess the quality that the prediction system will give us, since this kind of emergency may threaten urban areas and even human lives.

Based on the two-stage prediction framework, some previous studies were carried out in order to choose the most suitable adjustment strategy [23, 24]. GAs turned out to be the most appropriate technique among other Global Optimization Algorithms (such as *Tabu Search* and *Simulated Annealing*,) not only for their outstanding results but also because their nature favors parallel computing. The use of such an iterative adjustment technique implies that it leads us to the desired solution progressively, that is, the more the iterations we are able to perform, the better the solution we will be able to find. Obviously, this fact has a direct impact on the time incurred in the prediction process. So, in order to reach a good trade-off between quality and urgency, one must consider three main interrelated issues.

(a) The quality of the prediction is directly related to the quality of the adjustment, and the quality of the adjustment depends on how many times the adjustment technique is iterated, as well as on the number of scenarios tested per iteration.

(b) The amount of computing resources determines the amount of simulations that can be simultaneously executed per iteration at the adjustment stage.

(c) The response time in providing a prediction is a critical point and seriously limits the number of iterations that can be executed at the adjustment stage.

The solution proposed is conceived to fulfill the necessity of deploying a way to set up in advance:

(i) the prediction scheme settings, in particular the adjustment policy's specific parameters, for a required quality of the prediction; this is especially relevant when the ongoing fire may threaten urban areas and even human lives;

(ii) the computational resources needed to deliver a required quality of the prediction, given certain time constraints.

In order to properly tackle this objective, the details of the proposed methodology for the characterization of the

Table 1: Dependencies between each factor belonging to the two-stage prediction framework.

| Simulation | | Adjustment | | Final prediction | |
|---|---|---|---|---|---|
| Time | Quality | Time | Quality | Time | Quality |
| Input settings<br>Computational resources | Input settings | Simulation time<br>Computational resources<br>Configuration of the adjustment method | Adjustment time | Simulation time | Adjustment quality and real-time eventualities |

two-stage prediction method are given in the subsequent subsection.

*2.1. Two-Stage Prediction Method Characterization.* Based on the assumptions and requirements detailed above, we have designed a methodology to characterize the two-stage prediction process. For this methodology to be as flexible as possible (simulator-independent), we have to analyze the behavior of the aforementioned processes in terms of the main variables that we must deal with: the time spent and the quality of the results. Table 1 summarizes the dependencies for each case. As one can see, there is a series of dependencies from the prediction quality to the simulation time.

As regards the time needed for the final prediction process, it consists of a single simulation of the winning input setting at the adjustment stage, and the quality of this simulation is directly correlated to the quality obtained at the end of the adjustment process [25].

Since the developed adjustment methods are all iterative, the quality obtained at the end of the adjustment stage presents a single dependence: the time available to perform it. Regarding the necessary time, this process presents a couple of dependencies; obviously, the time incurred in each simulation will determine the overall adjustment time, but it also depends on the specific configuration of the adjustment method itself. Since the adjustment strategies may be run in a parallel way, in many cases this configuration can take greater advantage of the available resources. Thus, the time incurred in the adjustment process will also depend on the available computational resources.

Finally, each simulation, in terms of time needed, is dependent on the input parameters that describe the scenario being simulated and the underlying computational resources where it is executed. The quality of each simulation will depend on the input parameters, since it is determined by the similarity between the simulation run using those input parameters and the actual evolution of the fire. Therefore, the characterization must be done from the simulation process to the prediction process, so that by means of characterizing the time incurred in the simulation process in terms of its inputs and the computational resources, we can reach the final prediction assessment.

To accomplish this goal, it is necessary to characterize the adjustment strategy in such a way that it is possible to determine beforehand the number of iterations and the number of scenarios per iteration that should be executed to ensure a certain prediction quality. Since each scenario implies one execution of the simulation kernel, it is also necessary to characterize this simulation kernel to estimate the time required to run each simulation. For this reason, we rely on a methodology for classifying the scenarios according to how long their simulation will take before their execution, based on the use of Decision Trees [26] as classification strategy. This methodology is widely discussed in previous works, such as [27, 28], and allows us to carry out a premature detection of lengthy simulations so that we can avoid their inclusion in the adjustment process. As it will be seen in Section 4, this represents a very important advantage for the success of our proposed methods.

Having characterized the adjustment technique and the simulation time, then it is possible to determine the necessary computing resources to execute a certain number of iterations with a certain number of simulations per iteration. Thus, our methodology allows us to determine the required computing resources to reach a certain prediction quality in a given time.

## 3. Genetic Algorithm Characterization

By their own nature, GAs constitute a technique that works in an iterative way; that is, the quality of its results directly depends on the times it is iterated as well as its specific configuration settings.

It starts with an initial population of individuals which will be evolved over several iterations in order to guide them to better search space areas. The individuals used in the case of forest fire spread prediction are defined as a sequence of different genes, namely, wind speed and wind direction, moisture content of the live fuel, moisture content of the dead fuel (at three different times), and type of vegetation, out of the 13 standard Northern Forest Fire Laboratory fuel models [29]. Topographic data is constant, so it is not considered in the evolutionary process.

Operators such as *elitism*, *selection*, *crossover,* and *mutation* are applied to every population to obtain a new one superior to the previous one. Elitism consists of keeping the best individuals from one generation to the next one. By this way, it is guaranteed that the solution coming from a certain generation is, at least, as good as the one obtained from the previous generation. Selection operation selects good quality parents to create children that will inherit their parents' good characteristics (by crossover operation). In order to guarantee natural diversity of individual characteristics, mutation can occur for each child characteristic (under a very slight probability).

Each individual from a population is ranked according to a predefined fitness function. Since simulated fires can be represented as a grid of cells map, indicating which cells were burned as a consequence of the simulated fire, in our case the quality of a specific individual is determined by means of the fitness function expressed by (1). This equation calculates the differences in the number of cells burned, both missing or in excess, between the simulated and the real fire. Formally, this formula corresponds to the *symmetric difference* between the actual spread and the simulated spread, divided by the actual spread, so as to express a proportion. *UnionCells* is the union of the number of cells burned in the real fire and the cells burned in the simulation, *IntersecCells* is the intersection between the number of cells burned in the real fire and in the simulation, *RealCells* are the cells burned in the real fire, and *InitCells* are the cells burned at the starting time:

$$\text{Error} = \frac{(\text{UnionCells} - \text{InitCells}) - (\text{IntersecCells} - \text{InitCells})}{\text{RealCells} - \text{InitCells}}. \quad (1)$$

The iterative nature of GAs leads to a near optimal solution in the adjustment stage after a certain number of iterations. For this reason, it is mandatory to analyze the GA convergence for the particular case of forest fire spread prediction, as well as to be able to extract a general characterization of its behavior.

The characterization of GAs as an adjustment technique for the two-stage prediction method must allow us to properly estimate the adjustment quality we may obtain, given certain restrictions, both in terms of timing and resource availability when the adjustment stage is done. Adjustment quality stands for the difference between the simulation result once the adjustment process is completely carried out and the real state of the hazard.

In general terms, this issue is addressed by means of performing a statistical analysis to determine, for each adjustment strategy, the features that affect the quality of the results. Then, a study of the particular impact of these factors on the convergence of each method is done, to infer criteria in order to estimate the achievable quality of results under certain restrictions.

Parameters, such as number of generations, individuals per population, elitism factor, and mutation probability, affect the quality of the winner individual, that is, the final solution we will deliver at the end of the adjustment process.

*3.1. GA Characterization Methodology.* Thus, our proposed methodology for the GA characterization can be summarized in the following steps.

(1) GA key settings identification: it is necessary to assess and determine the settings that could affect the quality of the results when using GA as the adjustment technique. So far, the analyzed factors in this work are the size of the populations, the number of generations, and the mutation probability.

(2) Study of the impact on convergence: depending on the specific case (mainly, the topographic area), the influence of certain GA settings upon the quality of the results may be decreased in favor of others and vice versa. Therefore, it is necessary to study the particular impact of these factors on the convergence for each case.

(3) Statistical analysis: this analysis consists of identifying which probability distribution best fits the results obtained, in terms of the quality of the adjustment obtained. Then, by means of the corresponding probability density function, we are able to establish a guarantee degree in our estimations, as well as determine which configuration of the GA is most suitable.

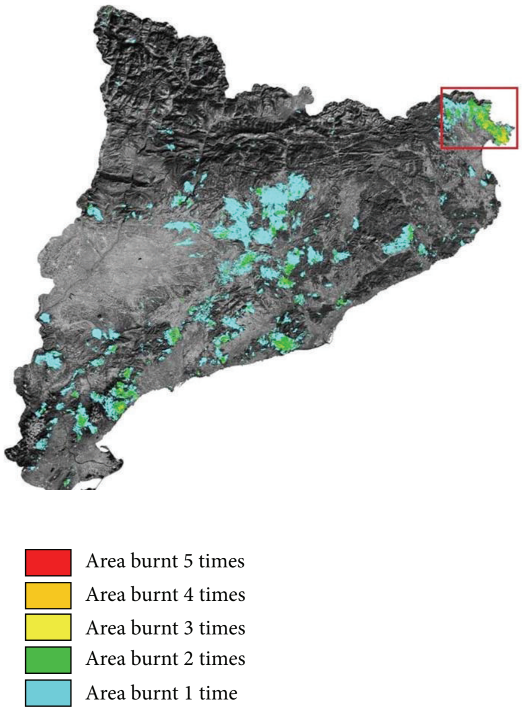


Figure 3: Fire occurrences in Catalonia in the period 1975–2010 (fire sizes bigger than or equal to 30 ha).

These steps are made *offline*, which means that they are all already carried out at the moment of the fire occurrence. It is also worth mentioning that the results and applicability of this methodology are independent of the features of the computational platforms being used.

Having completed these three processes, we are ready to infer criteria to determine the achievable adjustment quality under certain restrictions and are therefore able to advise on the best specific settings for the GA.

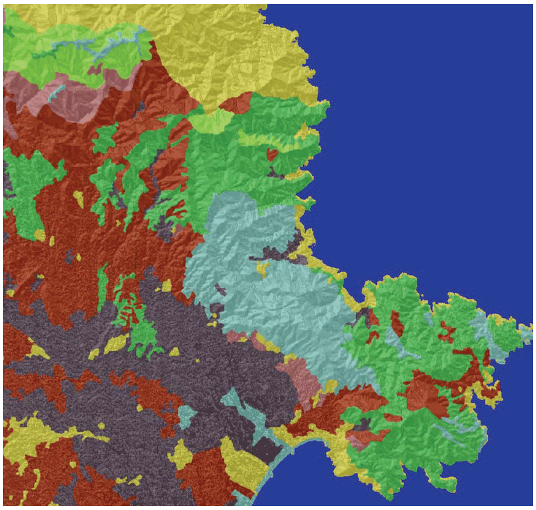


Figure 4: *Cap de Creus*, Catalonia (image from FARSITE simulator). Different colors indicate different types of vegetation.

## 4. Experimental Study: GA Characterization Based on *Cap de Creus* Landscape

The experimental study presented in this work is based on one of the most problematic areas in Catalonia (northeast of Spain). Figure 3 shows the occurrences of fires bigger than 30 hectares in the period 1975–2010 [2, 30]. The most recurrent area corresponds to the northeast cape (*El Cap de Creus*). This area is zoomed in in Figure 4 and has an approximate real extension of 900 square kilometers.

Subsequently, we test our methodology using this landscape. The fire spread simulator used was FARSITE [9].

*4.1. GA Statistical Study.* The analysis of the convergence and the statistical study were performed in the following terms:

(i) populations composed of 25 and 100 individuals;

(ii) populations evolved over 10 generations;

(iii) mutation probability: 10%;

(iv) crossover probability: 70%;

(v) initial fire consisting of a single initial ignition point;

(vi) adjustment time interval: 0 h–6 h (from the ignition to 6 hours after).

Using these configurations for the GA, we carried out the evolution process for 100 different populations.

From the obtained results, a statistical study carrying out the Kolmogorov-Smirnov, Anderson-Darling, and Chi-squared tests allowed us to determine that the probability distribution which best fits the obtained data, regarding the obtained quality of adjustment, is the *Lognormal* distribution, which is a continuous probability distribution of a random variable, whose logarithm is normally distributed. Its probability density function (PDF) is as follows:

$$\operatorname{PDF}(x;\mu,\sigma)=\frac{1}{x\sigma\sqrt{2\pi}}e^{-(\ln x-\mu)^2/2\sigma^2}\qquad x>0. \tag{2}$$

In this equation, $x$ is the random variable (which corresponds to the error obtained at the end of the adjustment process), $\mu$ is the mean, and $\sigma$ is the standard deviation of the variables natural logarithm (by definition, the variables logarithm is normally distributed).

Although the probability distribution of the data is the same throughout the whole evolution process, these factors vary depending on the iteration of the GA we are evaluating. So, Figure 5 depicts the different PDFs for each generation, for the particular case of 25-individual populations.

By means of these PDFs, we can guarantee the maximum adjustment error we will obtain, given a certain configuration of the GA, with different degrees of certainty. In addition, since the number of generations has a direct impact on both the available resources and the time needed to perform the adjustment process, it is worth highlighting that we are able to give this guarantee taking into account the number of generations we are able to execute.

For instance, Tables 2 and 3 show the different maximum adjustment errors (considering the adjustment time interval [0 hours–6 hours]) for which we have different *guarantee degrees* in the cases of 25 and 100 individuals per population, respectively. Here, *guarantee degree* stands for the probability of obtaining an adjustment error lesser than or equal to the specified value, on the basis of the above presented PDF (2).

Figures 6 and 7 also depict this information, from a guarantee degree of 90% down to 70%. As can be easily understood, the lesser the error requested is, the lesser the degree of guarantee, for the same number of iterations of the GA.

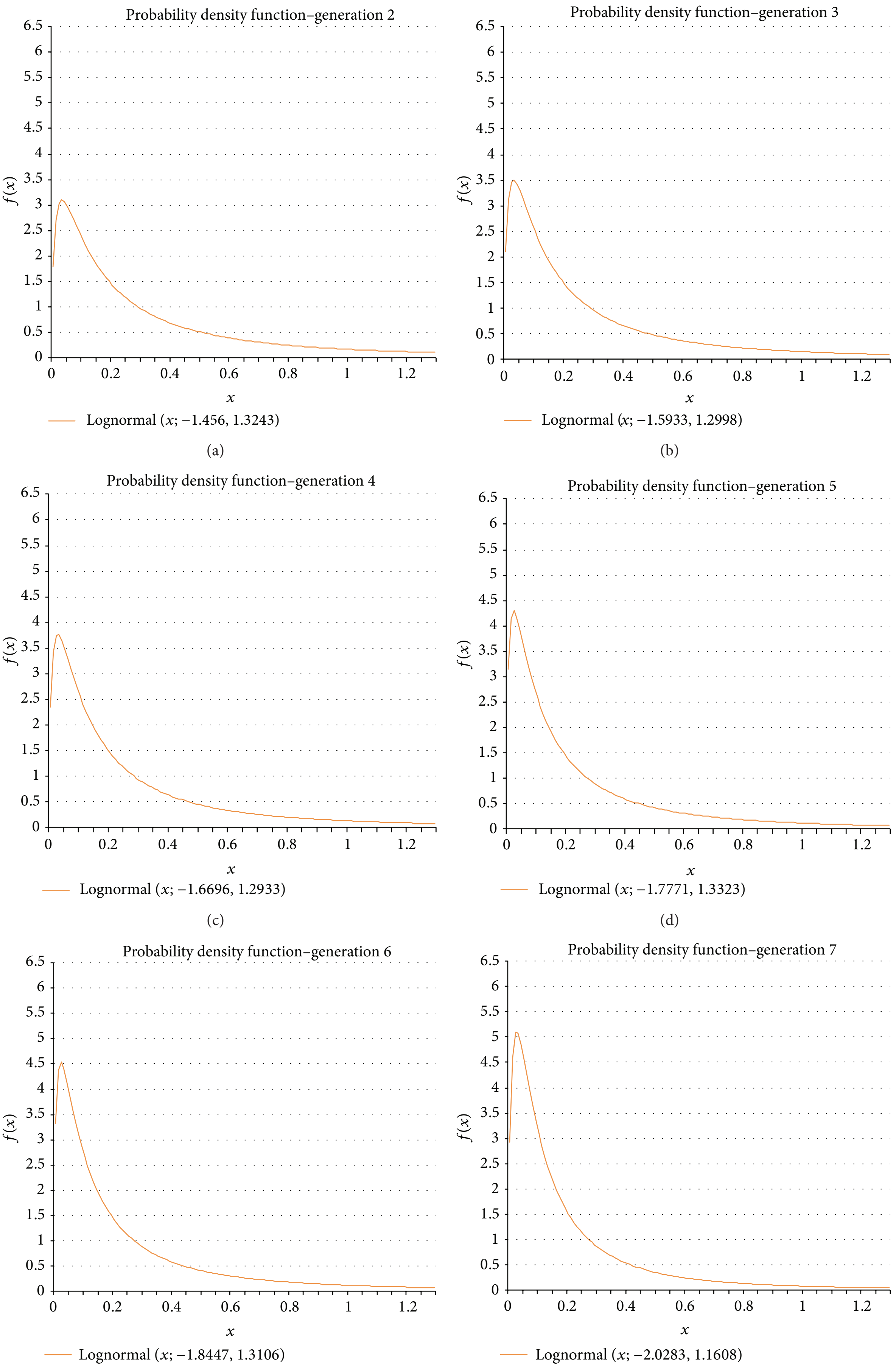
Considering a real situation, where the quality of the prediction is a parameter fixed by the decision control centre in charge of making the appropriate decisions, this information turns out to be very important, since we are able to give a certain guarantee of quality in the final prediction, taking into account how many evolution steps (i.e., how many generations) we can perform. Moreover, it is also possible to fix the quality of the adjustment and then determine the guarantee degree of reaching such an error in a given number of iterations.

*4.2. Real Emergency Recreation.* In order to prove the correctness of the above presented characterizations, we consider a hypothetical situation based on the *Cap de Creus* landscape, where we have to meet the following restrictions regarding both the quality of the adjustment and the time available to perform it.

(i) Response time: one hour for the adjustment process.

(ii) Quality of adjustment: a maximum error of approximately 0.5 is required.

As stated in Section 2.1, we rely on a Decision Tree-based method which allows us to rapidly assess the simulation time



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 5: Continued.

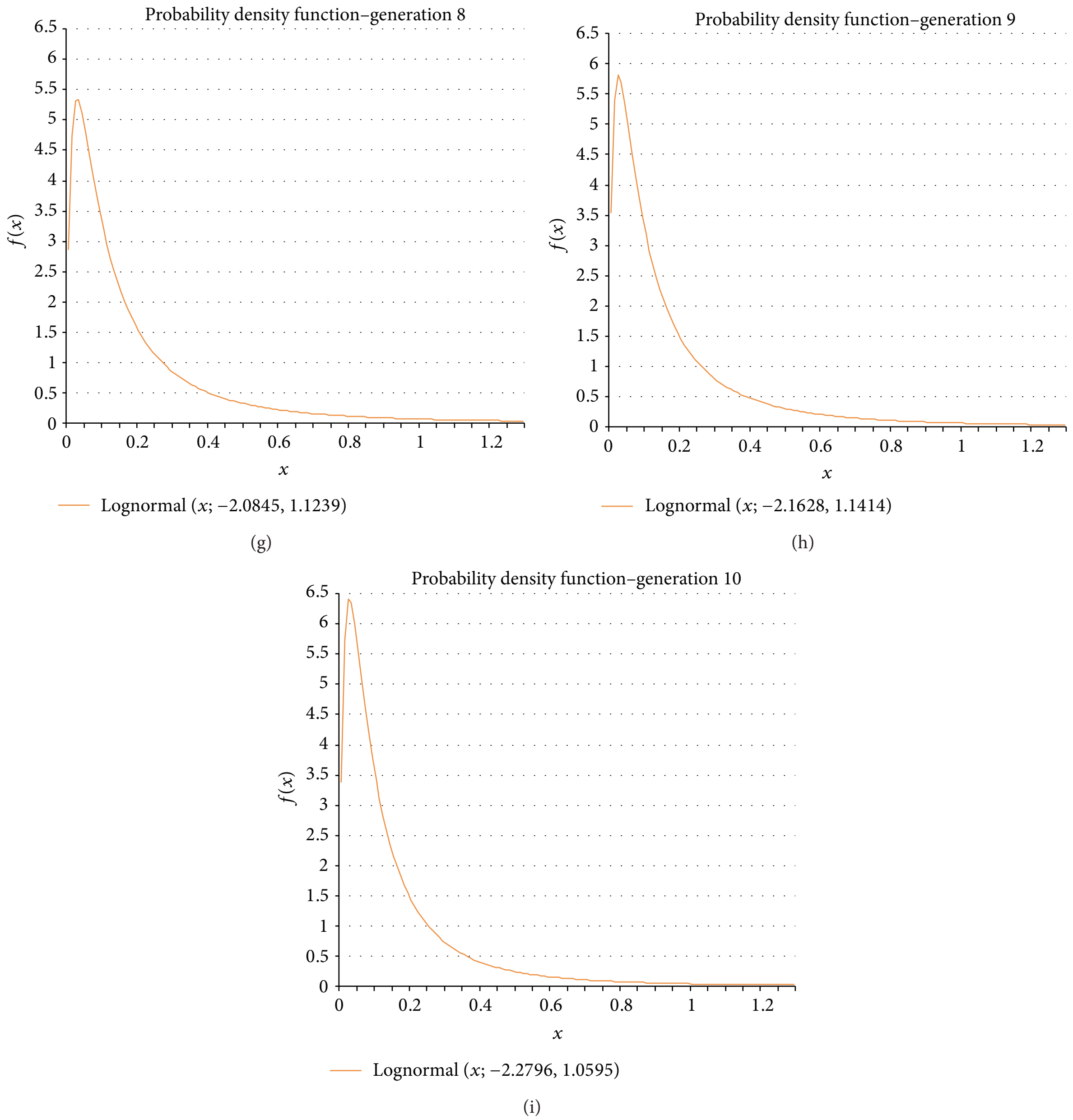


Figure 5: Probability density functions for the obtained errors at each generation of the evolution process.

a certain configuration of the scenario will produce, without the necessity of executing it [27, 28]. Therefore, by means of the premature detection and discard of lengthy simulations, we are able to limit the elapsed time in the adjustment stage.

For the particular case of this work, we applied this method to allow the inclusion of individuals whose simulations do not exceed 750 seconds (this detection process presented a hit ratio of 96.4% using a test set of 1000 individuals randomly generated). By this way, assuming that we have enough computational resources so as to execute 25 simulations in parallel (25 computing cores), we are able to evolve a population composed of 25 individuals over 5 generations in 3750 seconds (1 hour, 2 minutes, and 30 seconds), which would be appropriate.

By analyzing the data in Table 2, we assume that we could bring an error approximately equal to 0.5 with an 80% degree of guarantee at the end of the fifth generation (the error indicated is 0.519, row "80%", column "G5").

So, we carried out an experiment consisting of the evolution of 30 populations composed of 25 individuals. At the fifth generation, we should obtain an error equal to or lesser than 0.519 in approximately 80% of cases.

Table 4 shows the results obtained from these evolutions. These evolution processes were extended up to 10 generations for further analysis purposes. As can be seen, only 3 populations (p1, p19, and p22) out of 30 exceeded an error of 0.5 at the end of the fifth generation, which goes beyond the assumed 80% degree of guarantee. Moreover, by limiting the

Table 2: Achievable error with different degrees of guarantee (populations composed of 25 individuals).

| Guarantee degree | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|
| 95% | 2.06 | 1.72 | 1.58 | 1.51 | 1.36 | 0.888 | 0.79 | 0.752 | 0.585 |
| 90% | 1.27 | 1.08 | 0.987 | 0.933 | 0.848 | 0.582 | 0.525 | 0.497 | 0.398 |
| 85% | 0.92 | 0.782 | 0.719 | 0.673 | 0.615 | 0.438 | 0.399 | 0.357 | 0.307 |
| 80% | 0.711 | 0.607 | 0.559 | 0.519 | 0.476 | 0.349 | 0.32 | 0.301 | 0.25 |
| 75% | 0.57 | 0.488 | 0.45 | 0.415 | 0.383 | 0.288 | 0.265 | 0.248 | 0.209 |
| 70% | 0.467 | 0.402 | 0.371 | 0.34 | 0.314 | 0.242 | 0.224 | 0.209 | 0.178 |
| 65% | 0.388 | 0.335 | 0.31 | 0.283 | 0.262 | 0.206 | 0.192 | 0.179 | 0.154 |
| 60% | 0.326 | 0.283 | 0.261 | 0.237 | 0.22 | 0.177 | 0.165 | 0.154 | 0.134 |
| 55% | 0.275 | 0.239 | 0.222 | 0.2 | 0.186 | 0.152 | 0.143 | 0.133 | 0.117 |
| 50% | 0.233 | 0.203 | 0.188 | 0.169 | 0.158 | 0.132 | 0.124 | 0.115 | 0.102 |

Table 3: Achievable error with different degrees of guarantee (populations composed of 100 individuals).

| Guarantee degree | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|
| 95% | 0.552 | 0.325 | 0.289 | 0.211 | 0.164 | 0.164 | 0.138 | 0.121 | 0.113 |
| 90% | 0.399 | 0.251 | 0.226 | 0.172 | 0.136 | 0.134 | 0.114 | 0.101 | 0.095 |
| 85% | 0.32 | 0.212 | 0.191 | 0.149 | 0.12 | 0.116 | 0.1 | 0.09 | 0.085 |
| 80% | 0.269 | 0.185 | 0.168 | 0.133 | 0.109 | 0.104 | 0.09 | 0.082 | 0.077 |
| 75% | 0.232 | 0.164 | 0.15 | 0.121 | 0.099 | 0.095 | 0.083 | 0.076 | 0.071 |
| 70% | 0.203 | 0.148 | 0.135 | 0.111 | 0.092 | 0.087 | 0.077 | 0.07 | 0.066 |
| 65% | 0.179 | 0.134 | 0.123 | 0.103 | 0.086 | 0.08 | 0.071 | 0.066 | 0.062 |
| 60% | 0.159 | 0.122 | 0.113 | 0.095 | 0.08 | 0.074 | 0.067 | 0.062 | 0.058 |
| 55% | 0.142 | 0.112 | 0.103 | 0.089 | 0.075 | 0.069 | 0.062 | 0.058 | 0.055 |
| 50% | 0.127 | 0.102 | 0.095 | 0.082 | 0.071 | 0.064 | 0.058 | 0.055 | 0.052 |

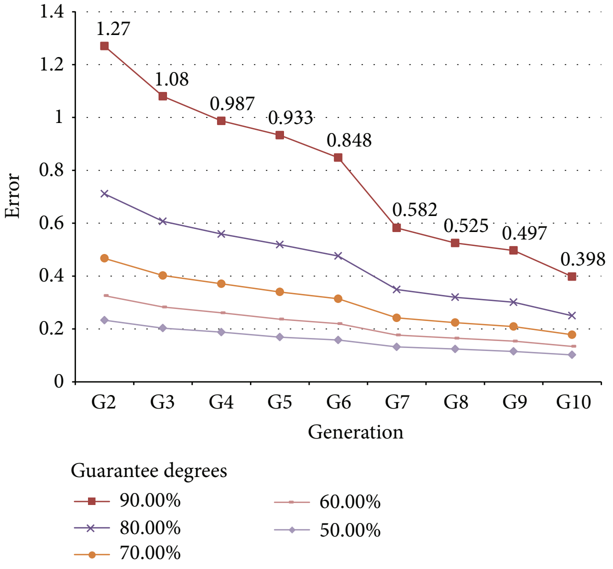


Figure 6: Achievable error with different degrees of guarantee. Populations composed of 25 individuals.

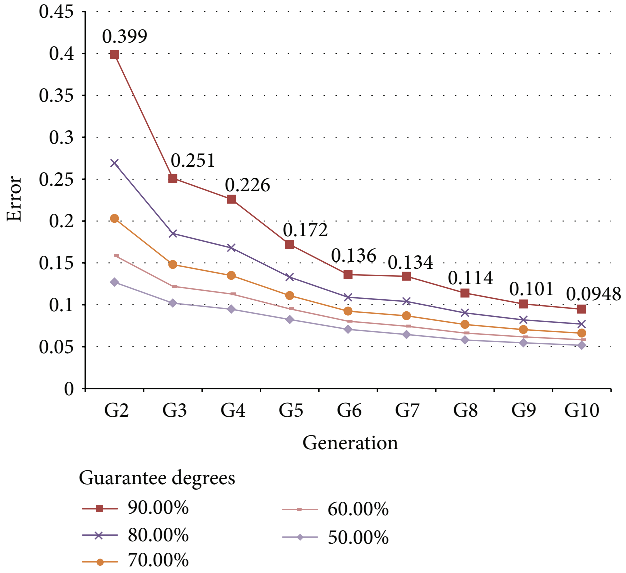


Figure 7: Achievable error with different degrees of guarantee. Populations composed of 100 individuals.

time involved in the adjustment process to 750 seconds, the response time restriction was also met.

Indeed, we can observe that in this experimental set only populations p19 and p22, in some generations, exceeded

Table 4: Errors obtained from the evolution of 30 populations composed of 25 individuals in *Cap de Creus* landscape.

| Population | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|
| p0 | 0.447 | 0.414 | 0.265 | 0.265 | 0.124 | 0.124 | 0.08 | 0.08 | 0.072 |
| p1 | 0.673 | 0.673 | 0.673 | 0.614 | 0.164 | 0.164 | 0.164 | 0.164 | 0.164 |
| p2 | 0.251 | 0.251 | 0.251 | 0.18 | 0.025 | 0.025 | 0.025 | 0.025 | 0.025 |
| p3 | 0.445 | 0.067 | 0.067 | 0.067 | 0.067 | 0.067 | 0.067 | 0.067 | 0.067 |
| p4 | 0.235 | 0.125 | 0.099 | 0.035 | 0.035 | 0.035 | 0.019 | 0.019 | 0.005 |
| p5 | 0.601 | 0.455 | 0.45 | 0.427 | 0.139 | 0.139 | 0.139 | 0.139 | 0.139 |
| p6 | 0.417 | 0.417 | 0.417 | 0.246 | 0.246 | 0.219 | 0.12 | 0.12 | 0.105 |
| p7 | 0.518 | 0.055 | 0.055 | 0.055 | 0.055 | 0.038 | 0.038 | 0.033 | 0.018 |
| p8 | 0.716 | 0.411 | 0.122 | 0.061 | 0.044 | 0.044 | 0.044 | 0.044 | 0.044 |
| p9 | 0.567 | 0.552 | 0.494 | 0.491 | 0.49 | 0.49 | 0.488 | 0.488 | 0.488 |
| p10 | 0.742 | 0.694 | 0.59 | 0.353 | 0.351 | 0.351 | 0.351 | 0.351 | 0.351 |
| p11 | 0.453 | 0.147 | 0.147 | 0.125 | 0.125 | 0.095 | 0.095 | 0.058 | 0.054 |
| p12 | 0.31 | 0.31 | 0.31 | 0.31 | 0.241 | 0.234 | 0.097 | 0.096 | 0.096 |
| p13 | 0.565 | 0.174 | 0.174 | 0.122 | 0.122 | 0.122 | 0.115 | 0.115 | 0.111 |
| p14 | 0.341 | 0.339 | 0.223 | 0.223 | 0.217 | 0.217 | 0.217 | 0.117 | 0.117 |
| p15 | 0.388 | 0.257 | 0.117 | 0.061 | 0.061 | 0.061 | 0.061 | 0.038 | 0.038 |
| p16 | 0.75 | 0.414 | 0.272 | 0.272 | 0.248 | 0.248 | 0.245 | 0.179 | 0.038 |
| p17 | 0.603 | 0.218 | 0.218 | 0.218 | 0.212 | 0.169 | 0.169 | 0.15 | 0.134 |
| p18 | 0.507 | 0.48 | 0.342 | 0.224 | 0.22 | 0.197 | 0.197 | 0.197 | 0.034 |
| p19 | 1.221 | 1.201 | 1.201 | 0.995 | 0.995 | 0.995 | 0.728 | 0.624 | 0.444 |
| p20 | 0.448 | 0.448 | 0.448 | 0.23 | 0.23 | 0.17 | 0.17 | 0.17 | 0.092 |
| p21 | 0.758 | 0.758 | 0.407 | 0.407 | 0.407 | 0.126 | 0.126 | 0.126 | 0.087 |
| p22 | 1.01 | 1.01 | 1.01 | 1.01 | 0.876 | 0.612 | 0.612 | 0.312 | 0.259 |
| p23 | 0.108 | 0.108 | 0.108 | 0.108 | 0.108 | 0.105 | 0.105 | 0.071 | 0.071 |
| p24 | 0.214 | 0.111 | 0.111 | 0.111 | 0.104 | 0.104 | 0.104 | 0.086 | 0.086 |
| p25 | 0.303 | 0.175 | 0.172 | 0.172 | 0.109 | 0.109 | 0.109 | 0.091 | 0.091 |
| p26 | 0.637 | 0.549 | 0.444 | 0.042 | 0.039 | 0.037 | 0.037 | 0.037 | 0.032 |
| p27 | 0.338 | 0.338 | 0.099 | 0.099 | 0.099 | 0.099 | 0.099 | 0.092 | 0.092 |
| p28 | 0.75 | 0.51 | 0.155 | 0.155 | 0.066 | 0.066 | 0.066 | 0.066 | 0.016 |
| p29 | 0.57 | 0.461 | 0.425 | 0.193 | 0.193 | 0.096 | 0.096 | 0.096 | 0.096 |

the errors corresponding to 90% degree of guarantee (see Table 2), which turns out to be an absolutely satisfactory result.

## 5. Conclusions

The novelty of our approach is the design of a methodology for the early assessment of both the time needed to perform a prediction and its quality within the proposed two-stage prediction framework, focusing on the specific case of forest fire spread prediction.

In the field of fire spread simulation and prediction, dealing with high degrees of uncertainty on the input parameters is common, which may lead to important losses in the quality of the predictions. The two-stage prediction scheme was introduced to relieve the problem of such input parameter uncertainty. This prediction framework highly improves the quality of the predictions but, due to its complexity, it could be very hard to know (or even to estimate) how much time is necessary in such processes, as well as the ideal amount and type of computational resources to be used.

Given the high level of uncertainty and complexity in the underlying scenario that describes the initial conditions of the catastrophe, it is necessary to rely on the so-called Global Optimization Algorithms to improve the quality of the predictions. The performance of these kinds of algorithms, however, may present high variability due to their specific initial settings. This variability may have impact on the quality of the results as well as the time needed to carry out the optimization process. For this reason, we present our methodology to characterize a well-known Artificial Intelligence technique, Genetic Algorithms, which has proven to be a powerful technique for performing the adjustment process in our two-stage prediction method. Thus, we relied on the probability theory area to carry out a statistical study based on a huge set of fire spread simulations using the *Cap de Creus* landscape as a real case study. Then, we have identified the probability distribution which corresponds to the results obtained, so that we can rely on its probability density function in order to establish certain degrees of guarantee in accuracy estimations.

This methodology allows us to keep to the occasional existing limitations as regards the time available to provide a prediction and the amount of computing resources we can access. This represents a great advantage for the forest fire operation management teams.

The experimental study presented highlights the suitability of the proposed methods, which can be easily extrapolated in order to be applied in management actions related to other kinds of natural hazards.

## Acknowledgment

This research has been supported by the MICINN-Spain under Contracts TIN2011-28689-C02-01 and TIN2007-64974.

## References

[1] European Environment Agency, "Mapping the impacts of natural hazards and technological accidents in Europe," Tech. Rep. 13, 2010.

[2] R. Salvador, R. Díaz-Delgado, J. Valeriano, and X. Pons, "Remote sensing of forest fires," in *Proceedings of GIS PlaNET'98 International Conference and Exhibition on Geographic Information (CD-ROM)*, 1998.

[3] J. Yang and Z. Fei, "HDAR: hole detection and adaptive geographic routing for ad hoc networks," in *Proceedings of the19th International Conference on Computer Communications and Networks (ICCCN '10)*, August 2010.

[4] B. Huang, A. J. Plaza, and Z. Wu, "High-Performance Computing in Remote Sensing III," in *Proceedings of SPIE*, vol. 8895, 2013.

[5] Y. S. Chang, D. Bao, G. X. Liu, W. Wu, and N. Ao, "The study of grassland fire loss assessment method based on remote sensing technology in inner Mongolia," in *Intelligent Systems and Decision Making for Risk Analysis and Crisis Response: Proceedings of the 4th International Conference on Risk Analysis and Crisis Response, Istanbul, Turkey, 27-29 August 2013*, CRC Press, 2013.

[6] R. Rothermel, "A mathematical model for predicting fire spread in wildland fuels," Tech. Rep. INT-115, USDA Forest Service, Ogden, Utah, USA, 1972.

[7] C. D. Bevins, "Firelib user manual and technical reference," 2008, http://www.fire.org/downloads/fireLib/1.0.4/doc.html.

[8] A. M. G. Lopes, M. G. Cruz, and D. X. Viegas, "Firestation—an integrated software system for the numerical simulation of fire spread on complex topography," *Environmental Modelling and Software*, vol. 17, no. 3, pp. 269–285, 2002.

[9] M. Finney, "FARSITE: Fire Area Simulator-model development and evaluation," Tech. Rep. RMRS-RP-4, US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, Utah, USA, 1998.

[10] G. L. Achtemeier, "Field validation of a free-agent cellular automata model of fire spread with fire-atmosphere coupling," *International Journal of Wildland Fire*, vol. 22, no. 2, pp. 148–156, 2012.

[11] E. Innocenti, X. Silvani, A. Muzy, and D. R. C. Hill, "A software framework for fine grain parallelization of cellular models with OpenMP: application to fire spread," *Environmental Modelling and Software*, vol. 24, no. 7, pp. 819–831, 2009.

[12] F. A. Sousa, R. J. N. dos Reis, and J. C. F. Pereira, "Simulation of surface fire fronts using fireLib and GPUs," *Environmental Modelling & Software*, vol. 38, pp. 167–177, 2012.

[13] B. Abdalhaq, A. Cortés, T. Margalef, and E. Luque, "Enhancing wildland fire prediction on cluster systems applying evolutionary optimization techniques," *Future Generation Computer Systems*, vol. 21, no. 1, pp. 61–67, 2005.

[14] H. Madsen and F. Jakobsen, "Cyclone induced storm surge and flood forecasting in the northern Bay of Bengal," *Coastal Engineering*, vol. 51, no. 4, pp. 277–296, 2004.

[15] S. D. Aberson, "Five-day tropical cyclone track forecasts in the North Atlantic basin," *Weather and Forecasting*, vol. 13, no. 4, pp. 1005–1015, 1998.

[16] H. Weber, "Hurricane track prediction using a statistical ensemble of numerical models," *Monthly Weather Review*, vol. 131, pp. 749–770, 2003.

[17] T. Weise, *Global Optimization Algorithms: Theory and Application*, Thomas Weise, 2nd edition, 2009.

[18] Y. Celik and E. Ulker, "An improved marriagein honey bees optimization algorithm for single objective unconstrained optimization," *The Scientific World Journal*, vol. 2013, Article ID 370172, 11 pages, 2013.

[19] S. Liu, H. Jin, X. Mao, B. Zhai, Y. Zhan, and X. Feng, "Selective segmentation for global optimization of depth estimation in complex scenes," *The Scientific World Journal*, vol. 2013, Article ID 868674, 9 pages, 2013.

[20] J. Qin, L. Ni, and F. Shi, "Combined simulated annealing algorithm for the discrete facility location problem," *The Scientific World Journal*, vol. 2012, Article ID 576392, 7 pages, 2012.

[21] K. Sheng Lim, Z. Ibrahim, S. Buyamin et al., "Improving vector evaluated particle swarm optimisation by incorporating nondominated solutions," *The Scientific World Journal*, vol. 2013, Article ID 510763, 19 pages, 2013.

[22] M. Denham, K. Wendt, G. Bianchini, A. Cortés, and T. Margalef, "Dynamic data-driven genetic algorithm for forest fire spread prediction," *Journal of Computational Science*, vol. 3, no. 5, pp. 398–404, 2012.

[23] B. Abdalhaq, A. Cortés, T. Margalef, and E. Luque, "Optimization of parameters in forest fire propagation models," Proceedings of the 4th International Conference on Forest Fire Research, Luso, Portugal, 2002.

[24] B. Abdalhaq, *A methodology to enhance the prediction of forest fire propagation [Ph.D. thesis]*, Universitat Autònoma de Barcelona, 2004.

[25] G. Bianchini, M. Denham, A. Cortés, T. Margalef, and E. Luque, "Wildland fire growth prediction method based on multiple overlapping solution," *Journal of Computational Science*, vol. 1, no. 4, pp. 229–237, 2010.

[26] T. M. Mitchell, *Machine Learning*, McGraw-Hill, New York, NY, USA, 1997.

[27] T. Artés, A. Cencerrado, A. Cortés, and T. Margalef, "Relieving the effects of uncertainty in forest fire spread prediciton by hybrid MPI-OpenMP parallel strategies," *Procedia Computer Science*, vol. 18, pp. 2278–2287, 2013.

[28] A. Cencerrado, R. Rodriguez, A. Cortés, and T. Margalef, "Urgency versus accuracy: dynamic data driven application system for natural hazard management," *International Journal of Numerical Analysis and Modeling*, vol. 9, no. 2, pp. 432–448, 2012.

[29] F. A. Albini, "Estimating wildfire behavior and effects," Tech. Rep. INT-30, US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, Ogden, Utah, USA, 1976.

[30] J. Burriel, F. Castro, T. Mata, D. Montserrat, E. G. Francisco, and J. Ibañez J, "La mejora del mapa diario de riesgo de incendio forestal en Cataluña," in *El acceso a la información espacial y las nuevas tecnologías geográficas*, pp. 651–666, 2006.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 202683, 10 pages
http://dx.doi.org/10.1155/2013/202683

*Research Article*

# An Improved Hierarchical Genetic Algorithm for Sheet Cutting Scheduling with Process Constraints

**Yunqing Rao,[1] Dezhong Qi,[1] and Jinling Li[2]**

[1] *The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science & Technology, Wuhan, Hubei 430074, China*
[2] *Shenyang Donfon Titanium Industry Co., Ltd, Shenyang, Liaoning 110168, China*

Correspondence should be addressed to Dezhong Qi; derek@hust.edu.cn

Received 24 September 2013; Accepted 18 November 2013

Academic Editors: T. Chen, Q. Cheng, and J. Yang

For the first time, an improved hierarchical genetic algorithm for sheet cutting problem which involves $n$ cutting patterns for $m$ non-identical parallel machines with process constraints has been proposed in the integrated cutting stock model. The objective of the cutting scheduling problem is minimizing the weighted completed time. A mathematical model for this problem is presented, an improved hierarchical genetic algorithm (ant colony—hierarchical genetic algorithm) is developed for better solution, and a hierarchical coding method is used based on the characteristics of the problem. Furthermore, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm. The computational result and comparison prove that the presented approach is quite effective for the considered problem.

## 1. Introduction

The sheet cutting is widely used in engineering machinery, mining machinery, port machinery, and other industry machinery. The manufacturing model and management of modern enterprises have been changed greatly by the development of science and technology. Since integrated cutting stock has become a new cutting model, the integrated cutting stock brings much economic benefits to the enterprise, but it also brings some difficulty for solving the nesting and cutting scheduling problem in the meantime. Therefore, finding an advanced approach to solving the cutting scheduling problem for integrated cutting stock has an important practical and theoretical significance.

The processing capacity of different types of cutting machines is different. For example, 0.15 mm~6 mm thick steel plate can be cut on the laser cutting machine, and the cutting speed can reach 10000 mm/min. Meanwhile, 5 mm~200 mm thick steel plate can be cut on the flame cutting machine, and its cutting speed is 0–700 mm/min. Therefore, the cutting patterns of different material and thickness can be cut on the different types of cutting machines. In other words, there are some process constraints in the process of integrated cutting stock. Furthermore, the same cutting pattern can be cut on the different types of cutting machines with different cutting speed.

Assignment of $n$ cutting patterns to $m$ unit different cutting machines is considered as an integrated cutting scheduling problem. Each cutting pattern can be only cut one time on a machine. The objective of integrated cutting scheduling is minimizing the weighted completed time. It belongs to scheduling problem for nonidentical parallel machines with process constraint and sequence-independent setups.

Lots of algorithms and approaches are used in cutting stock problem including conventional optimization algorithms and various metaheuristic algorithms. For instance, Blazewicz et al. [1], Gomes and Oliveira[2], Umetani et al. [3], Bennell et al. [4], Cui et al. [5], Cui and Chen [6], Xie et al. [7], and others proposed some nesting algorithms and gave some suggestions on the two-dimensional cutting stock problem. However, literatures on the cutting scheduling problem

(CSP) are not vast. There are some papers concerning the combined cutting stock and lot-sizing problem. NonÅs and Thorstenson [8] wanted to find an optimal production schedule involves the solution of a combined two-dimensional irregular cutting-stock and lot-sizing problem. In that paper, they proposed a problem formulation and suggested different solution algorithms (e.g., local search algorithms and simple tree-search algorithm) for a combined cutting-stock and lot-sizing problem. Gramani and França [9] analyzed the trade-off that arises when they solve the cutting stock problem by taking into account the production planning for various periods. In that paper, they formulated a mixed-integer mathematical model that combines the cutting stock and lot-sizing problems and developed a solution method based on an analogy with the network shortest path problem. NonÅs and Thorstenson [8] suggested a new column generating solution procedure (CGSP) for a combined cutting-stock and lot-sizing problem. Those papers are mainly concerned how to generate the cutting patterns, and minimize the trim-lost or trim-lost, setup cost and holding cost. However, how to assign those cutting patterns to different cutting machines, which belongs to cutting scheduling problem, is not concerned. As a good cutting scheduling can reduce production costs and raise the production efficiency, this problem is also important in the whole sheet cutting process. In this paper, our goal is to find a suitable cutting schedule and minimize the weighted completed time.

The cutting scheduling problem belongs to scheduling problem for nonidentical parallel machines with process constraint and sequence-independent setups. Scheduling problem has also been studied extensively, for example, Allahverdi et al. classified scheduling problems into batch and non-batch, sequence-independent and sequence-dependent setup, and categorizes the literature according to the shop environments of single machine, parallel machines, flow-shops, and job shops [10]. Each scheduling problem is denoted by the standard threefield notation $\alpha/\beta/\gamma$. The first field $\alpha$ describes the scheduling type, the second field $\beta$ is reserved for the information and conditions of scheduling, while the third field $\gamma$ contains the performance criteria. Cheng and Sin surveyed the major research results in deterministic parallel-machine scheduling [11]. And lots of the literatures of scheduling have considered parallel-machine scheduling problems. Peng and Liu, Anglani et al., Fowler et al., and other scholars have done some works on parallel machines problem [12–14]. And the vast majority of these studies have concentrated on studying the case of identical parallel machines. However, the case of nonidentical parallel-machine schedules has more practical sense than the case of identical parallel machines in real production (e.g. cutting and scheduling problem belong to nonidentical parallel-machine schedule).

Literatures on nonidentical parallel-machine schedules problem are not vast. Li and Yang gave a review of the nonidentical parallel-machine total weighted/weighted completion time problems [15]. Van Hop and Nagarur proposed a new approach to solve the PCB scheduling problem on a set of nonidentical machines. This approach model which related tasks of grouping, sequencing, and component switching as one integrated problem, with an objective of minimizing the total makespan [16]. Balin proposed a GA approach, which minimized maximum completion time (makespan) and considered nonidentical parallel machine scheduling problem with fuzzy processing times [17]. In order to adapt GA to nonidentical parallel machine scheduling problem, he proposed a new crossover operator and a new optimality criterion. Alcan and Balişgil presented a kind of genetic algorithm based on machine code for minimizing the processing times in nonidentical machine scheduling problem [18]. Also triangular fuzzy processing times were used in order to adapt the GA to nonidentical parallel machine scheduling problem in that paper. Besides the above mentioned papers, there are a few others that have investigated the unrelated-parallel machine scheduling problems with different approach (Liaw et al. [19], Rocha et al. [20], Mehravaran and Logendran [21], Arnaout et al. [22], Charalambous and Fleszar [23]). However, even though there are some papers considered nonidentical parallel-machine scheduling, few of them deal with nonidentical parallel machine with process constraints.

Taken together, how to assign cutting patterns to different cutting machines (i.e., the sheet cutting scheduling problem) is a key point for the whole sheet cutting process, but this issue are not involved in the previous papers. Considering the practical requirement, some further work on the sheet cutting problem needs to be done. To solving this problem, a mathematical model which takes into account the weighted completed time is presented. Furthermore, an improved hierarchical genetic algorithm (ant colony—hierarchical genetic algorithm) is developed to solve this mathematical model. Based on the characteristics of the problem, a hierarchical coding method is used in this algorithm. In addition, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm.

The rest of the paper is organized as follows. In Section 2, a detailed description of the sheet cutting scheduling problem is illustrated. A mathematic model for the sheet cutting problem is presented in Section 3. In Section 4, a process for solving the sheet cutting problem with a hierarchical genetic algorithm is demonstrated. Then the results of computational experiment are described in Section 5. Finally, conclusions and directions are given in Section 6.

## 2. Statement of the Problem

The process of sheet cutting is as follows. First, a variety of cutting patterns with the combination of different parts are generated by special nesting software. Second, those cutting patterns are assigned to different cutting machines. Thirdly, those cutting patterns are cut by different cutting machines. The process of sheet cutting can be showed by Figure 1. As the processing capacity of different types of cutting machines is different, a cutting pattern can be cut on some special cutting machines under different cutting speed. The capacity of some different cutting machines is shown in Table 1.

For example, a cutting pattern, material Q235 and thickness 20 mm, can be cut on plasma cutting machine B, flame

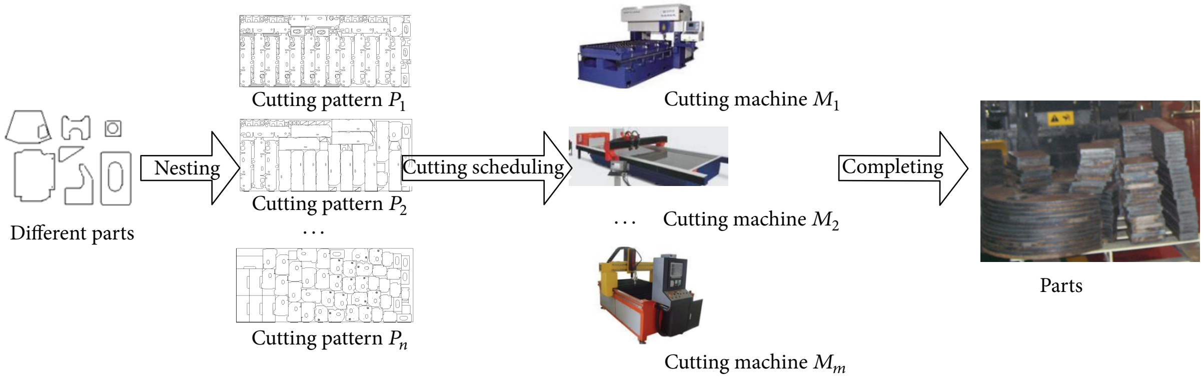


Figure 1: The production process of sheet cutting.

Table 1: The capacity of different cutting machines.

| Machine no. | Cutting machine type | The range of sheet thickness can be processed | Cutting speed |
|---|---|---|---|
| $M_1$ | Laser cutting machine | 0.15 mm~6 mm carbon steel | 10000 mm/min |
| $M_2$ | Plasma cutting machine A | 1 mm~15 mm carbon steel | 1000 mm/min |
| $M_3$ | Plasma cutting machine B | 1 mm~25 mm carbon steel | 800 mm/min |
| $M_4$ | Flame cutting machine A | 6 mm~200 mm carbon steel | 350 mm/min |
| $M_5$ | Flame cutting machine B | 6 mm~200 mm carbon steel | 450 mm/min |
| $M_6$ | Flame cutting machine C | 6 mm~200 mm carbon steel | 500 mm/min |

cutting machine A, flame cutting machine B and flame cutting machine C. The cutting speed is 800 mm/min on the plasma cutting machine B, but it is 500 mm/min on the flame cutting machine C. Therefore, how to assign cutting patterns to different cutting machines is needed to be solved by some method.

## 3. Mathematical Modeling

We consider the cutting scheduling problem of $n$ cutting patterns for $m$ nonidentical parallel machines with process constraints in detail, where $m < n$. $Tp_{ij}$ represents the processing time of cutting patterns, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$. It means the processing time of cutting pattern $P_i$ on the cutting machine $M_j$. Each cutting pattern has a corresponding weight coefficient $\omega_i$. We need to make an optimal scheduling to minimize the weighted completion time.

The problem considered can be summarized by the following points.

(1) $n$ different cutting patterns $P_1, P_2, \ldots, P_n$ need to be cut.

(2) Each cutting pattern includes some information such as number of parts, cutting length, number of punch, and so forth.

(3) $m$ different cutting machines $M_1, M_2, \ldots, M_n$ can be used. Each cutting machine can cut a variety of sheets of different material and thickness.

(4) The process time of a cutting pattern on different cutting machine may be different.

(5) The products pass rate is 100%, in other words, there is no reprocessing.

(6) Once starting cutting, it is not allowed to interrupt.

(7) Each cutting pattern is independent.

(8) Only one working procedure (i.e., cutting procedure) is included in this sheet cutting process.

(9) The cutting process includes special process constraint, which the cutting machine set $CP_i$ means cutting patterns $P_i$ can be used, $CP_i \in M$, $CP_i \neq \Phi$.

(10) Setup time includes punch time, collection time (the time of collect parts after completing cut) and the time using to adjust steel on the cutting machine.

(11) Empty travel time is ignored, in other words, the processing time is the actual cutting time.

*Definition.* Consider the following:

$n$ is the number of cutting patterns,

$m$ is the number of cutting machines,

$i$ is cutting pattern index, $i = 1, \ldots, n$,

$j$ is cutting machine index, $j = 1, \ldots, m$,

$pn_i$ is number of parts in cutting pattern $P_i$,

$ph_i$ is number of punch in cutting pattern $P_i$,

$$tp_{ij}\begin{cases}\text{cutting time of cutting pattern } i \text{ on machine } j\\ \text{if cutting pattern } P_i \text{ can be}\\ \text{processed on machine } j\\ \infty \text{ if cutting pattern } P_i \text{ cannot be}\\ \text{processed on machine } j,\end{cases} \tag{1}$$

$Vpm_{ij}$ is cutting speed of cutting pattern $P_i$ on cutting machine $M_j$,

$stp_{ij}$ is setup time of cutting pattern $P_i$ on cutting machine $M_j$,

$\omega_i$ is process weight of cutting pattern $P_i$,

$c_{ij}$ is completed time of cutting pattern $P_i$ on machine $M_j$.

*Decision Variables*. Consider the following:

$$z_{ij}\begin{cases}1 \quad \text{if cutting pattern } P_i\\ \text{can be cut on cutting machine } M_j\\ 0 \quad \text{otherwise}\end{cases} \tag{2}$$

*Objective Function*. Consider the following:

$$f = \min \sum_{i=1}^{n} \sum_{j,j\in CP_i} \omega_i c_{ij} z_{ij} \tag{3}$$

subject to

$$\sum_{j,j\in CP_i} z_{ij} = 1, \quad i = 1,2,\ldots,n \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j,j\in CP_i} z_{ij} = n, \quad i = 1,2,\ldots n, \ j = 1,2,\ldots m \tag{5}$$

$$c_{ij} = \left(c_{xj} + tp_{ij} + stp_{ij}\right) \times z_{ij}, \tag{6}$$

$$stp_{ij} = ph_i \times 1 + pn_i \times 0.5 + 5, \tag{7}$$

$$tp_{ik} \times Vmp_{ik} = t_{il} \times Vmp_{il} \quad k,l = 1,2,\ldots,m, \tag{8}$$

$$CP_i \in M, \quad CP_i \notin \Phi. \tag{9}$$

The minimizing weighted completed time can be got by objective function (3). Equation (4) considers a cutting pattern can be cut only on one cutting machine. All of cutting patterns are sure to be cut by (5). Equation (6) represents the completed time of cutting pattern $P_i$ on machine $M_j$. It consists of three components which involve $c_{xj}$, $t_{pj}$ and $stp_{ij}$, where $c_{xj}$ represents the completed time of cutting pattern $P_x$ which is cut on the cutting machine $M_j$ before cutting pattern $P_i$. There are three components for (7). The first component represents punch time (punch time of each drilling is 1 minute). The second component represents collection time (the time of collect each part which a cutting pattern includes cut is 0.5 minute). The last component relates to the time using to adjust steel on the cutting machine. Processing time of a cutting pattern $P_i$ on different machines can be expressed by (8). Formula (9) considers the cutting process contains a special process constraint. In other words, a cutting pattern is assigned which cutting machine depends on the sheet characteristics.

With the increase of the problem size, the solution of the problem becomes very complicated, or it is even impossible to be solved with conventional optimization methods. And it has been proved it is a NP-problem. In this paper, an ant colony—hierarchical genetic algorithm is considered to solve this problem. A set of optimized initial solution is generated by ant colony algorithm which has some merits, such as simple and universal, robustness. However, it is easy to fall into local optimum. Then, the hierarchical genetic algorithm which has strongly global search capability is used to further optimize the initial solution. Fast and efficient global optimizing can come true by using the ant colony—hierarchical genetic algorithm.

## 4. Ant Colony—Hierarchical Genetic Algorithm

A set of optimized initial solution is generated by using the ant colony algorithm for its simple and universal [24]. Then the solution is furthermore optimized by hierarchical genetic algorithm for its strongly global search capability. A hierarchical structure, which is consisted of parameter genes and control genes, is used by the hierarchical genetic algorithm [25]. The parameter genes are the lowest level, and control genes are in the higher levels of the parameter genes. The parameter genes are controlled by the control genes. In addition, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm.

*4.1. Code.* If the binary encoding is used for solving the cutting scheduling problem, the chromosome will be very complex and difficult for the crossover operator and decoding. So, to overcome these difficulties, the natural number encoding couple with binary encoding is used.

For example, six cutting patterns are assigned to three different cutting machines. The cutting sequence is showed by control genes and the cutting machine of select is showed by parameter genes. The value of parameter gene is 1 means the selected cutting machine is $M_1$. As is shown in Figure 2, the sequence of cutting patterns is $P_4, P_5, P_1, P_6, P_2$, and $P_3$ and the cutting pattern $P_1, P_2, P_3, P_4, P_5$, and $P_6$ are, respectively, assigned to $M_1, M_2, M_1, M_1, M_3$, and $M_2$. Each parameter gene can be further expressed by 0-1 variables $z_{ij}$.

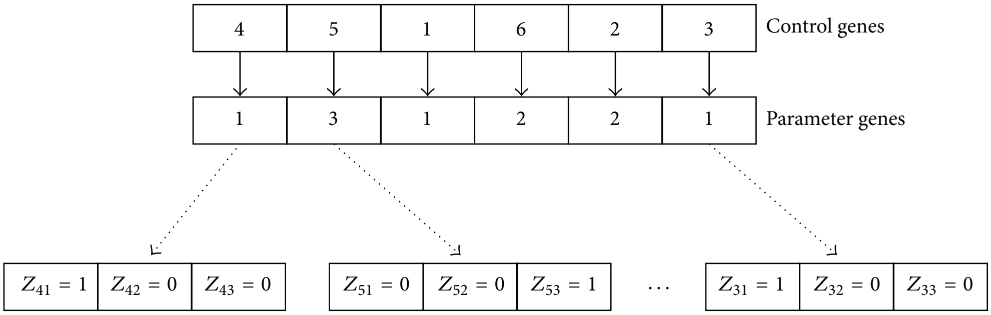


Figure 2: The process of natural number encoding.

*4.2. Initial Population.* The initial population is generated by ant colony algorithm. The following gives the concrete solving process of ant colony algorithm.

*Selecting the Ants' Path.* It supposes that the ants movement between nodes which represent cutting patterns on different cutting machines and different amount of pheromone left on the nodes at the same time. Then the pheromone impacts the path of the next lot size ants moving [26]. $\tau_{z_{ij}}(t)$ represents the pheromone values at $t$ ($t = 0, 1, 2, \ldots$) moment on the different nodes. The initial of the pheromone value $\tau_{z_{ij}}(0) = \varepsilon$ (the $\varepsilon$ is a minimal number). There are $N_{\text{ant}}$ ants distributed on the different nodes. Then each ant according to the pheromone value of next node and the heuristic factor independently chooses the next node. $p^k_{z_{ij}z_{(i+1)x}}(t)$ represents that the transition probability of ant $k$ ($k = 1, 2, \ldots, N_{\text{ant}}$) moves from the node $z_{ij}$ to the next node $z_{(i+1)x}$ at time $t$. The formulation of $p^k_{z_{ij}z_{(i+1)x}}(t)$ can be described as follows:

$$p^k_{z_{ij}z_{(i+1)x}}(t) = \begin{cases} \dfrac{\left[\tau_{[z_{ij}][z_{(i+1)x}]}(t)\right]^{\alpha}\left[\eta_{[z_{ij}][z_{(i+1)x}]}(t)\right]^{\beta}}{\sum_{x,x\in \text{CP}_{(i+1)}}\left[\tau_{(i+1)x}(t)\right]^{\alpha}\left[\eta_{(i+1)x}(t)\right]^{\beta}} \\ z_{(i+1)x} \notin \text{tabu}_k, \quad x \in \text{CP}_{(i+1)} \\ 0 \qquad \text{otherwise,} \end{cases} \tag{10}$$

where $\text{tabu}_k$ represents the next node set which ant $k$ cannot go to. $\alpha$ is a positive parameter, whose value represents the relative influence of pheromone trail. It shows the relative importance of ant track. The bigger the value of $\alpha$ is, the ant more inclines to choose the path which others have passed. $\beta$ is a positive parameter, whose value represents the relative influence of heuristic information. The value of $\beta$ is bigger, the state transition probability will be close to the greed rules. $\eta_{[z_{ij}][z_{(i+1)x}]}(t)$ is a heuristic function, whose value represents the expected next node ($z_{(i+1)x}$) what the ant expects to choose. The value of $\eta_{[z_{ij}][z_{(i+1)x}]}(t)$ can be gained by (11), where $c_{(i+1)x}$ represents the completion time of next node:

$$\eta_{[z_{ij}][z_{(i+1)x}]}(t) = 1 - \frac{c_{(i+1)x}}{\sum_{x,x\in \text{CP}_{(i+1)}} c_{(i+1)x}}. \tag{11}$$

*Pheromone Updating.* Pheromone evaporation is inevitable; meanwhile, the ants deposit pheromone in each iteration. So the pheromone value $\tau_{z_{ij}}(t)$ is changing in each iteration. The pheromone value $\tau_{z_{ij}}(t)$ can be updated as following equations:

$$\begin{aligned}\tau_{[z_{ij}][z_{(i+1)x}]}&(t+1) \\ &= (1-\rho)\,\tau_{[z_{ij}][z_{(i+1)x}]}(t) + \Delta\tau_{[z_{ij}][z_{(i+1)x}]},(t)\end{aligned} \tag{12}$$

$$\Delta\tau_{[z_{ij}][z_{(i+1)x}]}(t) = \sum_{k=1}^{N_{\text{ant}}} \Delta\tau^{k}{}_{[z_{ij}][z_{(i+1)x}]}, \tag{13}$$

$$\begin{aligned}&\Delta\tau^{k}{}_{[z_{ij}][z_{(i+1)x}]} \\ &\quad = \begin{cases} \dfrac{Q}{Z_k} & \text{if ant } k \text{ travels on edge } \left(z_{ij}, z_{(i+1)x}\right) \\ 0 & \text{otherwise,} \end{cases}\end{aligned} \tag{14}$$

where $\rho$ ($0 < \rho < 1$) is the rate of pheromone evaporation, and $1 - \rho$ is the rate of pheromone retention. Reducing the pheromone values enables the algorithm to forget bad decisions made in previous iterations [27]. Thus, the pheromone updating can help ants to explore new area in the search space. $\Delta\tau^k_{[z_{ij}][z_{(i+1)x}]}$ represents the pheromone value deposited on edge ($z_{ij}$, $z_{(i+1)x}$) by ant $k$ at $t$ iteration; $\Delta\tau_{[z_{ij}][z_{(i+1)x}]}$ is the sum of the pheromone value deposited on edge ($z_{ij}$, $z_{(i+1)x}$) by all of ants; $Q$ represents pheromone strength that affects the convergence speed of the algorithm. $Z_k$ represents the objective function value of ant $k$ in this iteration.

The step by step details are given as shown in Algorithm 1.

*4.3. Fit Function.* Minimizing weighted completion time is objective function, and the fitness function can be obtained by the exponential transform of objective function:

$$f = a\exp\left(-b\sum_{i=1}^{n} w_{ij}c_{ij}z_{ij}\right), \tag{15}$$

```
Step1. Initialization
    Set ACO parameters $\alpha$, $\beta$, $\rho$, $\tau_{y_{jm}}$ (0), $NC_{\max}$ (Maximum iterations), $Q$, $N_{\text{ant}}$ (number of ant);
Step2. Generate $N_{\text{ant}}$ ants, and place it on different nodes
Step3. for $NC = 1{:}NC_{\max}$
            for $k = 1{:}\ N_{\text{ant}}$
                for $i = 2{:}\ n$
                   Cumulating the transition probability of each ant will go to according (11), ants select the next node of max
        transition probability
                End;
            End;
Step4. Cumulating the best solution for the objective function (3) in this iteration. If the current solution is better than the
        former best solution, then update the former best solution.
Step5. updating the pheromone value according to (12), (13) and (14).
Step6.      if $NC < NC_{\max}$
                $NC = NC + 1$;
              Go to Step 2;
            Else
            Output the optimal solution;
            End;
        End;
```

Algorithm 1

where $a$ is a positive real number, $b$ is obtained by formula (18)

$$b = \begin{cases} N \times M \times \dfrac{f_{\min}}{f_{\min} + f_{\max}} & f \geq f_{\text{avg}} \\ N \times M \times \dfrac{f_{\max}}{f_{\min} + f_{\max}} & f \leq f_{\text{avg}}, \end{cases} \tag{16}$$

where $N$ is evolution generation, $M$ is the number of individuals in the population, $f_{\text{ave}}$ is the average fitness, $f_{\min}$ is the minimum fitness of the individual, $f_{\max}$ is the maximum fitness of the individual, and $f$ is the individual fitness.

*4.4. Selection.* The hierarchical genetic algorithm allows the population to progress from one generation into the next. The selection process is based on the fitness of the individuals, higher fitness results in more frequent selection. There are different selection rules such as the roulette wheel implementation, tournament selection, and elitism. Roulette wheel selection method is used.

Firstly, the fitness of individual $i(f_i)$ is calculated by formula (15), then the selected probability of individual $i$ can be calculated by the following formula:

$$P_i = \frac{f_i}{\sum_{k=1}^{n} f_k}. \tag{17}$$

Secondly, the cumulative probability of each chromosome is calculated by formula (20):

$$q_i = \sum_{i=1}^{l} P_i, \tag{18}$$

where $l$ represents the iteration times.

Final, using roulette selection method selects the individual.

*4.5. Crossover.* Some of the genetic material of two individuals are swapped (i.e., crossover operator), creating new individuals (children), who are possibly better than their parents. There are different crossover operator such as mapping crossover, different location crossover, same location crossover and leading crossover. For control gene of chromosome, partially mapping crossover method is used. The process of partially mapping crossover is that firstly, selecting two crossover points from parents' chromosome; secondly, the fragment of parents chromosome between the two crossover points is exchanged; thirdly, for the other genes, if the genes do not belong to the exchanged fragment of parents chromosome, the genes retain their value, otherwise, the value of those genes can be got using partially mapping method. For example, two parent individuals is $p_1 = [6\ 5\ 3\ 1\ 2\ 4]$ and $p_2 = [5\ 1\ 2\ 6\ 4\ 3]$, if the crossover point is 2 and 4, the offspring of individuals is $q_1 = [5\ 1\ 2\ 6\ 3\ 4]$ and $q_2 = [2\ 5\ 3\ 1\ 6\ 4]$. The process of crossover operator is shown by Figure 3. The illegal individuals can be avoided with this method.

As the cutting scheduling problem involves process constraints, the crossover operator cannot be used for the parameter genes of chromosome. For example, the cutting pattern $P_1$ and $P_2$ are respectively assigned to the cutting machine $M_5$ and $M_1$, after some crossover operator, the cutting pattern $P_1$ and $P_2$ may be, respectively, assigned to the cutting machine $M_1$ and $M_5$, but the cutting pattern $P_1$ cannot be cut on cutting machine $M_1$. So some illegal individuals can be generated by using crossover operator for the parameter genes of chromosome.

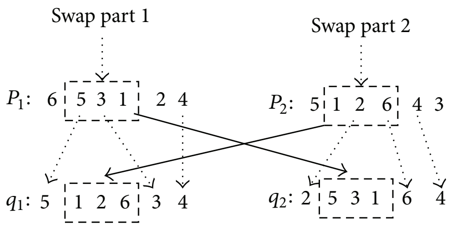


FIGURE 3: The process of crossover operator for control gene.

An adaptive process dynamically adjusts the operator's probabilities during the process of evolving a solution [28]. In order to accelerate evolutional speed and enlarge searching scope, an adaptive crossover probability $p_{\text{cross}}$ is designed:

$$p_{\text{cross}} = \begin{cases} p_{\text{cross}} \times \dfrac{1}{1 - f_{\min}/f_{\max}} & \text{if } \dfrac{f_{\min}}{f_{\max}} > e \\ p_{\text{cross}} & \text{others,} \end{cases} \quad (19)$$

where $f_{\text{ave}}$ is the individual's average fitness of each generation, $f_{\min}$ is the individual's minimum fitness of each generation, $f_{\max}$ is the individual's maximum fitness of each generation. $f_{\min}/f_{\max}$ is a positive parameter, whose value reflects the concentration of the whole generation. The bigger the value of $f_{\min}/f_{\max}$ is, this algorithm is more likely to fall into local optimal solution. If the the value of $f_{\min}/f_{\max}$ exceed a previously set threshold value $e$ $(0 < e < 1)$, the individuals tend to concentration.

*4.6. Mutation.* In order to explore new areas of the search space, the mutation with introducing a variation in the population and avoid premature convergence is needed. There are two types of mutation operators: control genes mutation and parameter genes mutation. For control genes of chromosome, changing sequence variation is used. In other words, randomly two points of parent chromosome is selected, then the value of the selected point of parents chromosome is exchanged each other. For example, the parent chromosome is $p = [6\ 5\ 3\ 1\ 2\ 4]$, selected the variation point is 2 and 5, the offspring chromosome is $p = [6\ 2\ 3\ 1\ 5\ 4]$ after variation operator. For parameter genes of chromosome, the integer variation is used, that is, the parent parameter gene is replaced by an integer $k$ $(k \in CP_i)$ with a certain probability.

An adaptive mutation probability $p_{\text{muta}}$ is designed as follows:

$$p_{\text{muta}} = \begin{cases} p_{\text{muta}} \times \dfrac{1}{1 - f_{\min}/f_{\max}} & \text{if } \dfrac{f_{\min}}{f_{\max}} > r \\ p_{\text{muta}} & \text{others.} \end{cases} \quad (20)$$

*4.7. Stop Criteria.* In this paper, setting a maximum iteration number has been used, and the algorithm will stop when the iteration reaches the setting maximum iteration.

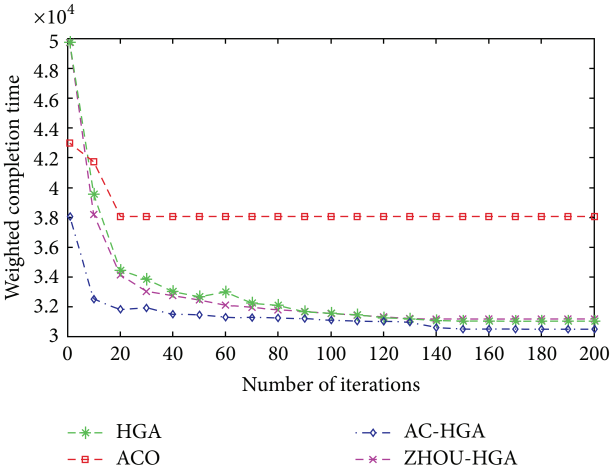


FIGURE 4: Average curve evolution.

## 5. Computational Experiments

In this section, the proposed optimization approach is proven to be available, and the performance of the solution strategy is evaluated by describing an experiment. First of all, a real set of cutting patterns, given by the metal forming factory, is tested by this approach. Later, the test results are reported and analyzed. The date of cutting patterns is given in Table 2, and the cutting machine date has been given in Table 1 in Section 2.

In this paper, the hierarchical genetic algorithm, ant colony algorithm and ant colony—hierarchical genetic algorithm are used for this experiment. Where, the initial population size of hierarchical genetic algorithm is 100; the maximum iterations number of hierarchical genetic algorithm is 200; the threshold value $e$ is 0.7; the threshold value $r$ is 0.8; the initial probability of control genes crossover $p_{\text{cross}}$ is 0.7, the initial probability of control genes mutation $p_{\text{muta}}$ is 0.6, the initial probability of parameter genes mutation is $p_{\text{muta}}$ 0.4; the information heuristic factor $\alpha$ of ant colony algorithm is 0.9; the expect heuristic factor $\beta$ of ant colony algorithm is 6; the pheromone evaporation rate $\rho$ is 0.3, the maximum iterations number of ant colony algorithm is 200, the initial population size of of ant colony algorithm is 30; the intensity of pheromone $Q$ is 1000.

The result of the example cutting scheduling problem with process constraints used different algorithms is given by Table 3. The evolution curve of different algorithms is also shown in Figure 4. Learned from Table 3, the result of using ant colony—hierarchical genetic algorithm is best. The minimizing weighted completion time is 30510 min. The Gantt chart of each cutting machine is shown in Figure 5.

## 6. Conclusion and Deductions

This paper discusses the scheduling problem of integrated cutting stock with process constraints. The objective of minimizing weighted completion time of $n$ cutting patterns for $m$ nonidentical parallel machines is considered. There are three

Table 2: The data of cutting patterns.

| Cutting pattern | Weight | Cutting length | Number of parts | Number of punch | Material | Thickness | Available cutting machine |
|---|---|---|---|---|---|---|---|
| $P_1$ | 5 | 43256 | 29 | 34 | Q235 | 8 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_2$ | 2 | 16656 | 13 | 16 | Q235 | 8 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_3$ | 4 | 12533 | 26 | 36 | Q235 | 8 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_4$ | 3 | 28768 | 22 | 22 | Q235 | 10 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_5$ | 5 | 11465 | 17 | 20 | Q235 | 10 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_6$ | 4 | 11909 | 22 | 29 | Q235 | 10 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_7$ | 3 | 118920 | 37 | 44 | Q235 | 10 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_8$ | 1 | 3763 | 5 | 8 | Q235 | 10 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_9$ | 5 | 32729 | 34 | 38 | Q235 | 12 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{10}$ | 3 | 17464 | 21 | 29 | Q235 | 12 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{11}$ | 4 | 18671 | 27 | 35 | Q235 | 12 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{12}$ | 4 | 72579 | 33 | 39 | Q345 | 15 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{13}$ | 5 | 58832 | 38 | 48 | Q345 | 15 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{14}$ | 4 | 218320 | 31 | 43 | Q345 | 15 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{15}$ | 1 | 13640 | 11 | 19 | Q345 | 15 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{16}$ | 3 | 11393 | 19 | 27 | Q345 | 15 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{17}$ | 5 | 106680 | 38 | 50 | Q345 | 15 | $M_2, M_3, M_4, M_5, M_6$ |
| $P_{18}$ | 5 | 5928 | 8 | 15 | Q345 | 20 | $M_3, M_4, M_5, M_6$ |
| $P_{19}$ | 3 | 9564 | 9 | 13 | Q345 | 20 | $M_3, M_4, M_5, M_6$ |
| $P_{20}$ | 5 | 83235 | 17 | 23 | Q345 | 20 | $M_3, M_4, M_5, M_6$ |
| $P_{21}$ | 1 | 29441 | 7 | 17 | Q345 | 20 | $M_3, M_4, M_5, M_6$ |
| $P_{22}$ | 2 | 133620 | 27 | 36 | Q345 | 20 | $M_3, M_4, M_5, M_6$ |
| $P_{23}$ | 5 | 71432 | 14 | 18 | Q345 | 24 | $M_3, M_4, M_5, M_6$ |
| $P_{24}$ | 1 | 162360 | 35 | 42 | Q345 | 24 | $M_3, M_4, M_5, M_6$ |
| $P_{25}$ | 1 | 69178 | 10 | 16 | Q345 | 24 | $M_3, M_4, M_5, M_6$ |
| $P_{26}$ | 2 | 85800 | 12 | 16 | Q345 | 24 | $M_3, M_4, M_5, M_6$ |
| $P_{27}$ | 1 | 125400 | 32 | 39 | Q345 | 30 | $M_4, M_5, M_6$ |
| $P_{28}$ | 4 | 159370 | 26 | 34 | Q345 | 30 | $M_4, M_5, M_6$ |
| $P_{29}$ | 2 | 144100 | 34 | 46 | Q345 | 32 | $M_4, M_5, M_6$ |
| $P_{30}$ | 1 | 57115 | 12 | 19 | Q345 | 32 | $M_4, M_5, M_6$ |

Table 3: The comparison of different algorithms.

| Optimization method | Optimum solution | Average value (20) | Iterative times |
|---|---|---|---|
| ACO | 38063 | 38063 | 200 |
| HGA | 31063 | 31078 | 200 |
| AC-HGA | 30510 | 30537 | 200 |
| ZHOU-HGA [25] | 31198 | 32007 | 200 |

tasks contributing to solving the cutting scheduling problem. Firstly, a mathematical model for the cutting scheduling problem with process constraints is proposed. Secondly, an ant colony—hierarchical genetic algorithm is designed to solve the mathematical model. In the proposed technique, a set of initial solution is generated by ant colony algorithm, and then an optimum solution is obtained by using selection, crossover, and mutation of the hierarchical genetic algorithm. Where, the control genes are used to determine the sequence of the cutting pattern, and the parameter genes are employed to identify the cutting pattern assigned on cutting machine. Finally, computational experiment is performed to evaluate the performance of the proposed algorithm. It is verified that the proposed method on cutting scheduling problem with process constraints is effective. In addition, as Figure 4 shown, there reaches a conclusion that the proposed method overweighs other methods. The result of cutting scheduling problem is given by using Gantt chart.

The cutting scheduling plays an important role in sheet cutting process. The existing literatures give a deep research

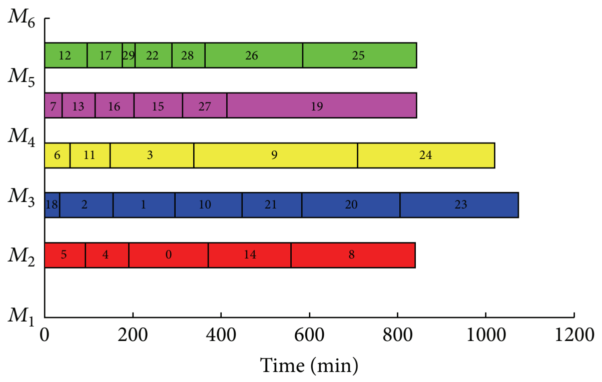


Figure 5: The Gantt chart of each cutting machine.

on cutting-stock problem and the combined cutting-stock and lot-sizing problem, but the study of the cutting scheduling problem has been ignored. To the best of our knowledge, the present paper is an effort to consider the cutting scheduling problem with process constraints for the first time, which involve $n$ cutting patterns for $m$ nonidentical parallel machines with process constraints. It belongs to scheduling problem for nonidentical parallel machines with process constraint and sequence-independent setups.

To solve this problem better, an improving hierarchical genetic algorithm is applied. A new encoding mode, that is, the natural number encoding couple with binary encoding, is adopted in this algorithm. In addition, to speed up convergence rates and resolve local convergence issues, a kind of adaptive crossover probability and mutation probability is used in this algorithm. Our approach for solving cutting scheduling problem has been applied in practical production process and has been accepted by sheet cutting manufacturers of china. This approach is novel and gives a different perspective for the manufacturing management of sheet cutting.

In this paper, the goal is focus on the minimizing weighted completion time. To achieve the target for solving the cutting scheduling problem, the minimum tardiness or other objectives can be considered. However, some products which needs sheet cutting do not merely includes cutting process (such as structural parts, after cutting process, its need to be bended, or other machined). In the future work, we will focus on the cutting scheduling problem with multi-processes.

## Acknowledgments

This research work is supported by the Science and Technology Major Project of China [Grant no. 2011ZX04015-011-07].

## References

[1] J. Blazewicz, V. Boljunčić, S. Martello, and J. Skorin-Kapov, "Combinatorial optimization issues in scheduling," *Journal of Scheduling*, vol. 14, no. 3, pp. 221–223, 2011.

[2] A. M. Gomes and J. F. Oliveira, "A 2-exchange heuristic for nesting problems," *European Journal of Operational Research*, vol. 141, no. 2, pp. 359–370, 2002.

[3] S. Umetani, M. Yagiura, and T. Ibaraki, "An LP-based local search to the one dimensional cutting stock problem using a given number of cutting patterns," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 5, pp. 1093–1102, 2003.

[4] J. A. Bennell, K. A. Dowsland, and W. B. Dowsland, "The irregular cutting-stock problem: a new procedure for deriving the no-fit polygon," *Computers and Operations Research*, vol. 28, no. 3, pp. 271–287, 2000.

[5] Y.D. Cui, J. L. Wu, and H. C. Chen, "Generating multi-section silicon steel sheet cutting patterns in the manufacturing industry of electric generators," *International Journal of Advanced Manufacturing Technology*, vol. 32, no. 3-4, pp. 310–314, 2007.

[6] Y. Cui and Q. Chen, "Simple heuristic for the constrained two-dimensional cutting problem," *Proceedings of the Institution of Mechanical Engineers B*, vol. 226, no. 3, pp. 565–572, 2012.

[7] S. Q. Xie, G. G. Wang, and Y. Liu, "Nesting of two-dimensional irregular parts: an integrated approach," *International Journal of Computer Integrated Manufacturing*, vol. 20, no. 8, pp. 741–756, 2007.

[8] S. L. NonÅs and A. Thorstenson, "Solving a combined cutting-stock and lot-sizing problem with a column generating procedure," *Computers and Operations Research*, vol. 35, no. 10, pp. 3371–3392, 2008.

[9] M. C. N. Gramani and P. M. França, "The combined cutting stock and lot-sizing problem in industrial processes," *European Journal of Operational Research*, vol. 174, no. 1, pp. 509–521, 2006.

[10] A. Allahverdi, J. N. D. Gupta, and T. Aldowaisan, "A review of scheduling research involving setup considerations," *Omega*, vol. 27, no. 2, pp. 219–239, 1999.

[11] T. C. E. Cheng and C. C. S. Sin, "A state-of-the-art review of parallel-machine scheduling research," *European Journal of Operational Research*, vol. 47, no. 3, pp. 271–292, 1990.

[12] J. Peng and B. Liu, "Parallel machine scheduling models with fuzzy processing times," *Information Sciences*, vol. 166, no. 1–4, pp. 49–66, 2004.

[13] A. Anglani, A. Grieco, E. Guerriero, and R. Musmanno, "Robust scheduling of parallel machines with sequence-dependent set-up costs," *European Journal of Operational Research*, vol. 161, no. 3, pp. 704–720, 2005.

[14] J. W. Fowler, S.-M. Horng, and J. K. Cochran, "A hybridized genetic algorithm to solve parallel machine scheduling problems with sequence dependent setups," *International Journal of Industrial Engineering*, vol. 10, no. 3, pp. 232–243, 2003.

[15] K. Li and S.-L. Yang, "Non-identical parallel-machine scheduling research with minimizing total weighted completion times: models, relaxations and algorithms," *Applied Mathematical Modelling*, vol. 33, no. 4, pp. 2145–2158, 2009.

[16] N. Van Hop and N. N. Nagarur, "The scheduling problem of PCBs for multiple non-identical parallel machines," *European Journal of Operational Research*, vol. 158, no. 3, pp. 577–594, 2004.

[17] S. Balin, "Non-identical parallel machine scheduling with fuzzy processing times using genetic algorithm and simulation," *International Journal of Advanced Manufacturing Technology*, vol. 61, no. 9–12, pp. 1115–1117, 2012.

[18] P. Alcan and H. Balişgil, "A genetic algorithm application using fuzzy processing times in non-identical parallel machine scheduling problem," *Advances in Engineering Software*, vol. 45, no. 1, pp. 272–280, 2012.

[19] C.-F. Liaw, Y.-K. Lin, C.-Y. Cheng, and M. Chen, "Scheduling unrelated parallel machines to minimize total weighted tardiness," *Computers and Operations Research*, vol. 30, no. 12, pp. 1777–1789, 2003.

[20] P. L. Rocha, M. G. Ravetti, G. R. Mateus, and P. M. Pardalos, "Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times," *Computers and Operations Research*, vol. 35, no. 4, pp. 1250–1264, 2008.

[21] Y. Mehravaran and R. Logendran, "Bicriteria supply chain scheduling on unrelated-parallel machines," *Journal of the Chinese Institute of Industrial Engineers*, vol. 28, no. 2, pp. 91–101, 2011.

[22] J. Arnaout, G. Rabadi, and H. Mun, "A dynamic heuristic for stochastic unrelated parallel machine scheduling problem," *International Journal of Operations Research*, vol. 3, no. 2, pp. 136–143, 2006.

[23] C. Charalambous and K. Fleszar, "Variable neighborhood descent for the unrelated parallel machine scheduling problem," *International Journal on Artificial Intelligence Tools*, vol. 21, no. 4, Article ID 1240019, 18 pages, 2012.

[24] X. W. Luo, "Study on the Ant colony optimization," *Key Engineering Materials*, vol. 467–469, pp. 300–305, 2011.

[25] H. Zhou, W. Tang, and B. Niu, "Optimization of multiple traveling salesman problem based on hierarchical genetic algorithm," *Application Research of Computers*, vol. 26, no. 10, pp. 3754–3757, 2009.

[26] H. Zhang, A. Li, and X. Liu, "Hybrid ant colony algorithm for TSP," *Computer Engineering*, vol. 35, no. 8, pp. 34–37, 2009.

[27] M. Mavrovouniotis and S. Yang, "Ant colony optimization with immigrants schemes in dynamic environments," in *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN XI '10)*, pp. 371–380, 2010.

[28] A. Tuson and P. Ross, "Adapting operator settings in genetic algorithms," *Evolutionary Computation*, vol. 6, no. 2, pp. 161–184, 1998.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 270249, 11 pages
http://dx.doi.org/10.1155/2013/270249

*Research Article*

# An Effective Hybrid Self-Adapting Differential Evolution Algorithm for the Joint Replenishment and Location-Inventory Problem in a Three-Level Supply Chain

**Lin Wang,[1] Hui Qu,[1] Tao Chen,[2] and Fang-Ping Yan[1]**

[1] *School of Management, Huazhong University of Science and Technology, Wuhan 430074, China*
[2] *College of Public Administration, Huazhong University of Science and Technology, Wuhan 430074, China*

Correspondence should be addressed to Tao Chen; chentao15@163.com

Received 29 September 2013; Accepted 27 October 2013

Academic Editors: T. Chen, Q. Cheng, and J. Yang

The integration with different decisions in the supply chain is a trend, since it can avoid the suboptimal decisions. In this paper, we provide an effective intelligent algorithm for a modified joint replenishment and location-inventory problem (JR-LIP). The problem of the JR-LIP is to determine the reasonable number and location of distribution centers (DCs), the assignment policy of customers, and the replenishment policy of DCs such that the overall cost is minimized. However, due to the JR-LIP's difficult mathematical properties, simple and effective solutions for this NP-hard problem have eluded researchers. To find an effective approach for the JR-LIP, a hybrid self-adapting differential evolution algorithm (HSDE) is designed. To verify the effectiveness of the HSDE, two intelligent algorithms that have been proven to be effective algorithms for the similar problems named genetic algorithm (GA) and hybrid DE (HDE) are chosen to compare with it. Comparative results of benchmark functions and randomly generated JR-LIPs show that HSDE outperforms GA and HDE. Moreover, a sensitive analysis of cost parameters reveals the useful managerial insight. All comparative results show that HSDE is more stable and robust in handling this complex problem especially for the large-scale problem.

## 1. Introduction

As a multiitem inventory replenishment policy, the joint replenishment (JR) which can save the total costs by grouping multiitems in the same order had received numerous attentions [1]. While research on the classic joint replenishment problem (JRP) reached a saturation point, many extension versions had been proposed. These extensions can be divided into two branches: (a) relax the assumptions (such as storage, demand, and capacity limitation) of the classic JRP to simulate more realistic problems. The extensive literature review is available in [1]; (b) integrate more supply chain decisions (strategic or operational).

Silva and Gao [2] pointed out that three important decisions, facility location decisions, inventory management decisions, and distribution decisions, are highly related within a supply chain. Although independent policy always leads to a degree of suboptimization [3], these decisions usually are undertaken separately due to the complexity of integration. The existing works of literature were mainly the any two of three decisions integration. Wang et al. [4] proposed a two-level joint replenishment and delivery problem (JRD) by extending the model of Qu et al. [5]. The comparative results verified that the hybrid differential evolution (HDE) algorithm is effective. Cha et al. [6] considered the JRD of a three-level supply chain. A heuristic algorithm and an intelligent algorithm were proposed to solve this JRD. An extension of this JRD by adopting the freight consolidation policy was proposed by Moon et al. [7]. All the papers mentioned above were the integration of inventory and distribution decisions within one warehouse or distribution center (DC). The cost-saving can be achieved by sharing fixed ordering cost and transportation cost and economies of scale. With the development of global purchasing and supply chain management, there had been a strong move towards integration of location and inventory decisions between multi-DC (Berman et al. [8]).

Except Berman et al. [8] and Silva and Gao [2], the early studies on location-inventory problems (LIPs) had almost exclusively assumed the continuous-review $(r, Q)$ inventory policy. Teo et al. [9] formulated a LIP ignoring the shipping costs from the DCs to retailers. Daskin et al. [10] studied a LIP including location cost, inventory related costs, transportation costs from the supplier to the DCs, and local delivery costs from the DCs to the retailers. They proposed a Lagrangian relaxation solution algorithm for the proposed model. Shen et al. [11] considered the similar problem of Daskin et al. [10] by restructuring the model as a set-covering model and used column generation to solve the model. Based on the work of Daskin et al. [10] and Shen et al. [11], several works of literature had been published on this topic [3, 12–14].

The periodic-review $(R, S)$ policy was introduced into the LIP by Berman et al. [8]. They pointed out that the $(R, S)$ policy is easier to coordinate than the $(r, Q)$ policy and integrating the JR policy is a possible direction for future research. Silva and Gao [2] firstly proposed a LIP by considering the JR policy simultaneously. They solved this model using a two-stage method, the first stage, a Greedy Randomize Adaptive Search Procedure (GRASP) was used to determine the location decision; the second stage solved the JRP corresponding to the locations defined in the first stage. The proposed method is more suitable for the cases of specified number of opened DCs. This is the only one paper about the joint replenishment and location-inventory problem (JR-LIP). Thus it is meaningful to extend the JR-LIP and provide a new effective solving approach.

JRPs and uncapacitated facility location problems had been proven to be the NP-hard problems and they were rather hard to find effective algorithms [15, 16]. Current solution approaches for LIPs included different heuristics based on the complex mathematical analysis of the models, that is, Lagrangian relaxation based solution [10, 12], column generation algorithm [3, 11], conic integer programming approach [17, 18], and greedy randomized adaptive search procedure [2]. However the problem becomes much more complex than traditional LIPs because of the introduction of the JR policy (the ordering frequency of each DC and the basic ordering time needed to determine). It is rather hard to solve this JR-LIP effectively by traditional approaches. Firstly, available heuristics are too problem-specific and rather difficult to design. There exists no versatility approaches. Secondly, the enumeration is inefficient. It may take years to find a solution for a large problem size. On the other hand, due to the advantage of ignoring the mathematical property to search the optimal solution by starting from a feasible solution, intelligent algorithms had grown quickly in handling JRPs and JRDs. Results of similar studies illustrated that genetic algorithms (GAs) and differential evolution algorithms (DEs) are comparable suitable approaches [4, 19–22]. Although the existing works of literature hinted the superior of ant colony optimization (ACO) in solving combinatorial optimization, Wang et al. [22] pointed out that ACO was inferior in convergence rate for the similar JRD and more difficult to be expanded to complex problems.

The aim of this study is to propose a new and effective approach to handle the modified JR-LIP model based on the study of Silva and Gao [2]. The differences between our model and Silva and Gao [2] are as follows: (I) we assume that the maximum number of DC is known (additional variables were added in our model), the objective is to determine the number of DC to be opened, the locations of these DC, the assignment of customers, the replenishment frequency of each DC, and the replenishment cycle time to minimize the total system cost; (II) in Silva and Gao [2], the optimal number of DCs is obtained through contrasting the total cost by ADDING (or DROPPING) DC one by one. This method is not suitable for large-scale problems. The proposed hybrid self-adapting DE (HSDE) can directly find the optimal number of DCs through an intelligent search in a given space and is relatively easier to extend for large-scale problems. In fact, the HSDE can overcome the disadvantage of one-to-one competition used by classic DE and can avoid the manual parameter testing of mutation factor and crossover factor. Results of benchmark functions tests and numerical examples show the robustness of the HSDE.

The rest of this paper is organized as follows. Section 2 proposes the mathematical model. In Section 3, a DE-based algorithm is proposed to solve the JR-LIP. Section 4 is numerical examples and results discussion. Section 5 contains the conclusions and future research directions.

## 2. The JR-LIP Model and Analysis

*2.1. Assumptions and Notations.* A three-level supply chain consisting of multidistribution centers (DCs), an outside supplier, and multicustomers is considered. The item is ordered and collected by the DCs and then is distributed to customers. The objective is to decide the following policy: (1) how many DCs should be opened, and where to locate them; (2) how the customers are assigned to appropriate DCs; (3) how many and when to order, to minimize the total cost. The JR-LIP model is studied based on the following assumptions.

(i) Demand rates and costs are known and constant.

(ii) Shortages are not allowed.

(iii) Replenishment lead time is constant.

(iv) Each customer is only assigned to one DC, while other DCs cannot serve it.

(v) There is no limitation for the capacity of storage and shipment.

Only one item is considered in our model, DCs replenish their demand jointly. DC $i$ replenishes its item at every integer multiple $k_i$ of the basic cycle time $T$ and then delivers them to customer $j$. Figure 1 gives a simple description about our model.

Notions used in the model are as follows:

$i$: index of DCs, $1 \le i \le n$;

$j$: index of customers, $1 \le j \le m$;

$l$: index of potential sites for DC, $1 \le l \le m$;

$D_j$: annual demand rate for the item per unit time at customer $j$;

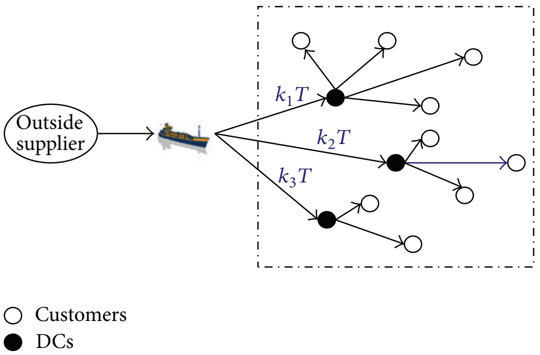


Figure 1: The three-level supply chain of proposed JR-LIP.

$S$: DCs' major ordering cost;

$s_l$: minor ordering cost of potential DC site $l$;

$h_l$: annual inventory holding cost of item at potential DC site $l$;

$c_{lj}$: the Euclidean distance between potential site $l$ and customer $j$;

$f_l$: the fixed cost of opening a DC at potential site $l$;

$T$: basic cycle time (decision variable);

$k_i$: ordering cycle time of item at DC $i$, integer number (decision variable);

$Z_i$: the variable to decide whether DC $i$ to be opened (decision variable), it is defined as

$$Z_i = \begin{cases} 1, & \text{if DC } i \text{ is opened,} \\ 0, & \text{otherwise;} \end{cases} \tag{1}$$

$Y_{il}$: the location variable (decision variable), it is defined as

$$Y_{il} = \begin{cases} 1, & \text{if DC } i \text{ is located at potential site } l, \\ 0, & \text{otherwise;} \end{cases} \tag{2}$$

$X_{ij}$: the assignment variable (decision variable), it is defined as

$$X_{ij} = \begin{cases} 1, & \text{if customer } j \text{ is assigned to DC } i, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

*2.2. Mathematical Model and Analysis.* The total cost is composed of the fixed location costs of DCs, assignment costs of DCs, and replenishment costs of DCs. For a given problem, we assume that the maximum number ($n$) of DCs is known and each site of customer is a potential site for a DC.

*(a) Location Costs.* The distance between customers is not considered in our model. Thus, besides the effect of replenishment policy, the main concern for making the location decision is the fixed opening cost and the distances from DCs to customers which are usually used to measure transportation costs in a supply chain network [2, 4]. So the total location costs ($C_L$) can be written as

$$C_L = \sum_{i=1}^{n}\sum_{l=1}^{m} f_l Y_{il} Z_i + \sum_{i=1}^{n}\sum_{l=1}^{m}\sum_{j=1}^{m} c_{lj} X_{ij} Y_{il} Z_i. \tag{4}$$

In (4), the first term is the fixed cost of locating DCs. The second term is the transportation cost of shipping from DCs to customers.

*(b) Replenishment Costs.* Similar to the classic JRP, the replenishment costs of DCs include ordering costs which consist of major ordering cost and minor ordering cost and inventory holding cost. The difference is the frequency of orders and the basic ordering cycle time at each DC is determined by the demand served by the DC. In turn, the demand served by the DC is a function of the assignment of customers to the DC. Thus the annual replenishment costs ($C_R$) are formulated as

$$C_R = \frac{S}{T} + \sum_{i=1}^{n}\sum_{l=1}^{m} \frac{s_l}{k_i T} Y_{il} Z_i + \sum_{i=1}^{n}\sum_{l=1}^{m} \frac{h_l k_i T D_i}{2} Y_{il} Z_i, \tag{5}$$

where $D_i = \sum_{j=1}^{m} D_j X_{ij}$ is the demand for the item by unit of time allocated to DC $i$. The first term is the major ordering cost of DCs; the second term is the total minor ordering cost of DCs; the third term is the annual inventory holding cost of DCs in (5).

*(c) The Objective.* According to the above analysis, the annual total cost (TC) is

$$\begin{aligned} \mathrm{TC} = C_L + C_R &= \sum_{i=1}^{n}\sum_{l=1}^{m} f_l Y_{il} Z_i + \sum_{i=1}^{n}\sum_{l=1}^{m}\sum_{j=1}^{m} c_{lj} X_{ij} Y_{il} Z_i \\ &+ \frac{S}{T} + \sum_{i=1}^{n}\sum_{l=1}^{m} \frac{s_l}{k_i T} Y_{il} Z_i + \sum_{i=1}^{n}\sum_{l=1}^{m} \frac{h_l k_i T D_i}{2} Y_{il} Z_i. \end{aligned} \tag{6}$$

The objective of JR-LIP model is

$$\begin{aligned} \min\ & \mathrm{TC}\left(X_{ij}, Y_{il}, Z_i, k_i, T\right) \\ & = \sum_{i=1}^{n}\sum_{l=1}^{m} f_l Y_{il} Z_i + \sum_{i=1}^{n}\sum_{l=1}^{m}\sum_{j=1}^{m} c_{lj} X_{ij} Y_{il} Z_i \\ & \quad + \frac{S}{T} + \sum_{i=1}^{n}\sum_{l=1}^{m} \frac{s_l}{k_i T} Y_{il} Z_i + \sum_{i=1}^{n}\sum_{l=1}^{m} \frac{h_l k_i T D_i}{2} Y_{il} Z_i \\ \text{s.t.}\ & \sum_{l=1}^{m} Y_{il} = 1, \quad \forall i, \\ & \sum_{i=1}^{n} X_{ij} = 1, \quad \forall j, \\ & X_{ij} - Z_i \le 0, \quad \forall i,\ \forall j, \\ & Y_{il} - Z_i \le 0, \quad \forall i,\ \forall l. \end{aligned} \tag{7}$$

The goal of this problem is to find the optimal $X_{ij}$, $Y_{il}$, $Z_i$, $k_i$, and $T$ to minimize the TC.

## 3. Problem Solving Methodology

Since all involved costs in the model are associated with the location of the DC, the thinking of solving methodology is converting (7) to a function with $Y_{il}$ firstly. Assuming that the index of opened DC ($Z_i$), the customers assigned to the opened DC ($X_{ij}$), and the replenishment policy of the opened DC ($k_i, T$) are known, the problem we faced is where to locate these opened DCs ($Y_{il}$) to minimize the total cost. For simplification, we denote that $\mathbf{A}$ is the set of opened DCs and $\mathbf{A}_i$ is the customer set of DCs $i$. Then the annual total cost can be rewritten as follows:

$$\begin{aligned} \mathrm{TC}\left(Y_{il}\right) = {} & \sum_{i\in A}\sum_{l=1}^{m} f_l Y_{il} + \sum_{i\in A}\sum_{l=1}^{m}\sum_{j\in A_i} c_{lj} Y_{il} + \frac{S}{T} \\ & + \sum_{i\in A}\sum_{l=1}^{m}\frac{s_l}{k_i T} Y_{il} + \frac{1}{2}\sum_{i\in A}\sum_{l=1}^{m} h_l k_i T D_i Y_{il}. \end{aligned} \tag{8}$$

For a given set $\{X, Z, k, T\}$, the total cost of each opened DC at each potential site ($\mathrm{TC}_{il}^{*}$) can be calculated easily. Sorting the total cost of each opened DC of all potential sites by ascending, the optimal $Y_{il}^{*}$ and $\mathrm{TC}_{il}^{*}$ of DC $i$ is the first site in the list. If there are $p$ DCs with the same best location index, the optimal $Y_{il}^{*}$ and $\mathrm{TC}_{il}^{*}$ is obtained by comparing at least $p$ combinations and at most $p!$ combinations from the first site to the $p$th site in the list. Finally, sum $\mathrm{TC}_{il}^{*}$ and the optimal total cost $\mathrm{TC}^{*}(Y_{il})$ are obtained.

### *3.1. The Proposed Hybrid Self-Adapting DE Algorithm (HSDE).*

Due to its easy implementation, quick convergence, and robustness, DE has turned to be one of the best evolutionary algorithms in a variety of fields [23–26]. However, the limitations on DE structure had inspired many scholars to improve upon DE by proposing modifications to the original DE fields [27].

*3.1.1. The works of Literature on DE's Modification.* Neri and Tirronen [28] made a comprehensive study on DE's recent advances. They used twenty-four benchmark functions to survey the performance of eight DE-based algorithms. The comparative results showed that self-adapting parameters of DE (jDE) proposed by Brest et al. [29] were the effective and most simple improvement. Furthermore this modification also can avoid the manual parameter setting of $F$ and CR.

The DE-based algorithms referred in Neri and Tirronen [28] were the modification for DE structure and many scholars also devoted to integrate other algorithm's superior scheme into DE. The most common integration is the combination of DE and GA operations. He et al. [30] combined GA and SQP operation into DE which is rather complex to be carried out for complex application cases. To avoid the limitation of one-to-one competition in DE, Lin [31] integrated the roulette wheel selection into DE. Further, Wang et al. [4] proposed a hybrid DE (HDE) which adopted a simpler selection scheme of GA named truncation selection. The numerical results showed that HDE is effective and can easily be applied in practical problems. The newest integration algorithm of DE that appeared recently is quantum-inspired differential evolution algorithms (QDEs) [32–35]. Since a quantum system with $n$ qubits can represent $2^n$ states simultaneously, the population size of the algorithm can be smaller, even one individual [36]. However, due to the complex encoding and decoding scheme, it was not easy for complex practical problems.

Table 1: HSDE's notations.

| Notation | Explanation |
|---|---|
| $N_p$ | Population: the number of individuals |
| $N_d$ | The dimension of the specific problem |
| $GenM$ | The maximum generation for evolution |
| $x_t^G$ | The target vector of individuals $t$ in $G$ generation |
| $v_t^G$ | The mutated vector of individuals $t$ in $G$ generation |
| $u_t^G$ | The trial vector of individuals $t$ in $G$ generation |
| $F_t^G$ | Mutation factor of individuals $t$ in $G$ generation |
| $\mathrm{CR}_t^G$ | Crossover factor of individuals $t$ in $G$ generation |

Based on the above analysis, we propose a DE-based algorithm named HSDE by integrating the GA and self-adapting parameters of DE.

*3.1.2. The Operations of HSDE.* The difference between HSDE and original DE is the structure of individual and selection operation. Table 1 lists the notations used in the intelligent algorithm. The details are discussed as follows.

*(1) Individual Structure.* In HSDE, the control parameters mutation factor and crossover factor are related to the individual and evolution generation, not constant for all individuals in the whole evolution process. Based on the study of Brest et al. [29], the individual structure of HSDE is

$$x_t^G = \left\{x_{t,1}^G, x_{t,2}^G, \ldots, x_{t,N_d}^G, F_t^G, \mathrm{CR}_t^G\right\}, \tag{9}$$

where $F_t^G$ and $\mathrm{CR}_t^G$ are updated by the following formulation:

$$F_t^G = \begin{cases} F_{\min} + \mathrm{rand}_1 \cdot F_{\max}, & \text{if } \mathrm{rand}_2 < \tau_1, \\ F_t^{G-1}, & \text{otherwise}, \end{cases} \tag{10}$$

$$\mathrm{CR}_t^G = \begin{cases} \mathrm{rand}_3, & \text{if } \mathrm{rand}_4 < \tau_2, \\ \mathrm{CR}_t^{G-1}, & \text{otherwise}, \end{cases} \tag{11}$$

where $\mathrm{rand}_1, \ldots, \mathrm{rand}_4$ are randomly generated number with uniform distributed between 0 and 1; $\tau_1$ and $\tau_2$ are constant values which represent the probabilities of parameters' update; $F_{\min}$ and $F_{\max}$ are the minimum and maximum $F$, respectively.

*(2) Mutation.* The mutated vector is obtained according to the following equation:

$$\begin{aligned} v_{t,1:N_d}^{G+1} = x_{r_1,1:N_d}^{G} + F_t^{G+1} * \left(x_{r_2,1:N_d}^{G} - x_{r_3,1:N_d}^{G}\right), \\ r_1 \neq r_2 \neq r_3 \neq t, \end{aligned} \tag{12}$$

where $F_t^{G+1}$ is obtained by (10); $r_1, r_2, r_3, t \in [1, 2, \ldots, N_p]$.

*(3) Crossover*. The crossover operation mixes the mutated vectors and the target vectors to increase the diversity of the parameter vector. The trial vector $u_t^{G+1} = \{u_{t1}^{G+1}, u_{t2}^{G+1}, \ldots, u_{tN_d}^{G+1}\}$ can be generated as:

$$u_{tq}^{G+1} = \begin{cases} v_{tq}^{G+1}, & \text{if rand}(q) \le \mathrm{CR}_t^{G+1} \text{ or } q = \text{rand}\,n(t), \\ x_{tq}^{G}, & \text{otherwise}, \end{cases} \tag{13}$$

where $q \in [1, 2, \ldots, N_d]$; rand$(q) \in [0, 1]$ is a randomly generated number with uniform distribution $\mathrm{CR}_t^{G+1}$ is obtained from (11); rand$\,n(t) \in [1, 2, \ldots, N_d]$ is a randomly selected integer to ensure that the trial vector gets at least one gene from mutated vector [37].

*(4) Selection*. Considering that the one-to-one competing scheme may obsolete the superior individual exclusively, the advantage of GA in selection is utilized. The modified selection scheme is as follows: generate a new population $\{x_1^G, x_2^G, \ldots, x_{N_p}^G, u_1^{G+1}, u_2^{G+1}, \ldots, u_{N_p}^{G+1}\}$ by combining the population $\{x_1^G, x_2^G, \ldots, x_{N_p}^G\}$ and $\{u_1^{G+1}, u_2^{G+1}, \ldots, u_{N_p}^{G+1}\}$; compute the fitness value of the new population; sort the fitness value obtained; select half individuals those fitness value at the front as the next generation population $\{x_1^{G+1}, x_2^{G+1}, \ldots, x_{N_p}^{G+1}\}$.

*3.2. The Procedures of HSDE for the JR-LIP.* Figure 2 shows the flow chart of HSDE-based procedures for the JR-LIP. Since the common operations of HSDE have been described in Section 3.1, we only focus on the procedures for the specific problem in this section, that is, initialization and decoding scheme, which are bolded in Figure 2.

*3.2.1. Initialization.* In the stage of initialization, we should determine the parameters of HSDE, the representation of chromosome, and the encoding and decoding scheme for the JR-LIP.

*The Parameters of HSDE.* Since the mutation factor and crossover factor for HSDE are encoded in the individual, the parameters that need to be determined are population size ($N_p$), the number of iterations (*GenM*), the lower bound ($F_{\min}$) and upper bound ($F_{\max}$) of mutation factor, and the probabilities of parameters' update ($\tau_1$, and $\tau_2$). As suggested in Brest et al. [29], we set $F_{\min} = 0.1$, $F_{\max} = 0.9$, and $\tau_1 = \tau_2 = 0.1$. The values of $N_p$ and *GenM* should be confirmed according to the practical problem as discussed in Section 4.

*The Representation of the Decoded Individual.* The proper representation of a solution plays a key role in the development of the HSDE. There are four parts included in the solution, the first part is the assignment information of customers; the second part is the replenishment frequency of DCs; the third part is the basic replenishment cycle time of DCs; the fourth part is the control parameters of HSDE. The dimension of the specific problem is $N_d = m + n + 1$ ($m$ customers' assignment information; $n$ DCs' ordering frequency; a basic ordering cycle time), and the total length of chromosome equals to $N_d + 2$. Figure 3 shows a decoded chromosome of twelve customers, three DCs.

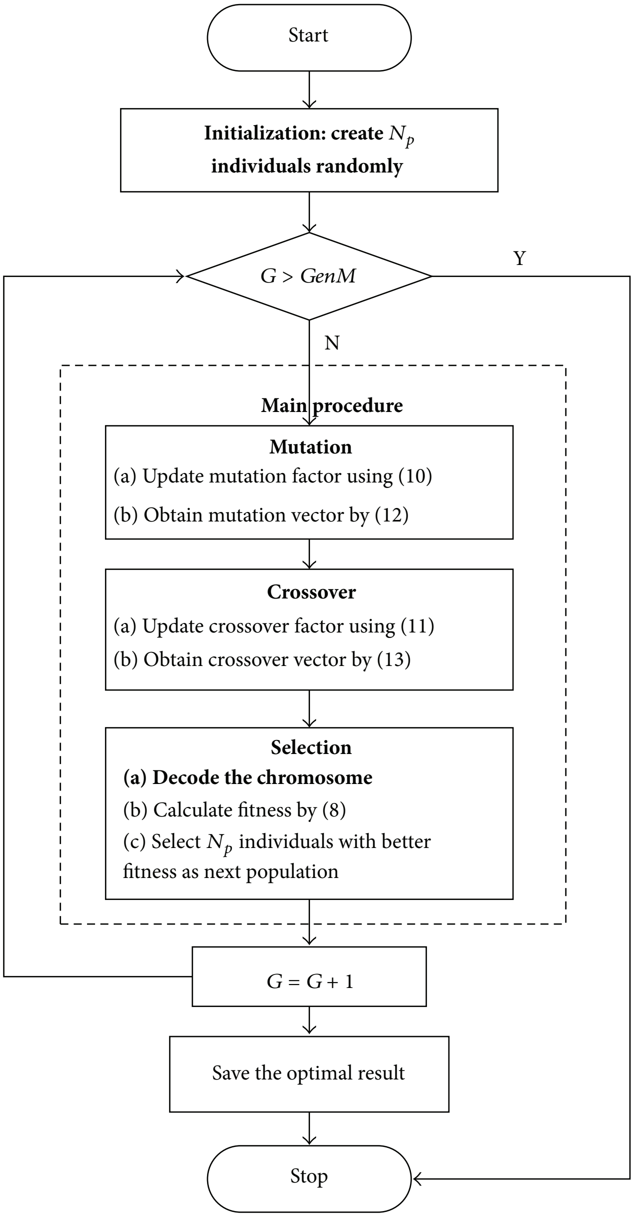
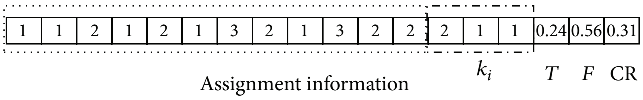

Start
Initialization: create $N_p$ individuals randomly
$G > GenM$
Y
N
Main procedure
Mutation
(a) Update mutation factor using (10)
(b) Obtain mutation vector by (12)
Crossover
(a) Update crossover factor using (11)
(b) Obtain crossover vector by (13)
Selection
(a) Decode the chromosome
(b) Calculate fitness by (8)
(c) Select $N_p$ individuals with better fitness as next population
$G = G + 1$
Save the optimal result
Stop

Figure 2: Flow chart of HSDE for the JR-LIP.



| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1 | 2 | 1 | 3 | 2 | 1 | 3 | 2 | 2 | 2 | 1 | 1 | 0.24 | 0.56 | 0.31 |

Figure 3: A decoded chromosome ($m = 12$, $n = 3$).

The first part of the decoded individual represents that customers 1, 2, 4, 6, and 9 are assigned to $\mathrm{DC}_1$; customers 3, 5, 8, 11, and 12 are assigned to $\mathrm{DC}_2$; customers 7, 10 are assigned to $\mathrm{DC}_3$. Thus, the decision variables $X_{ij}$ and $Z_i$ can be confirmed by assignment information. The rest of solution

Table 2: Benchmark functions.

| Test functions | $L$ | $M$ | Global optimum ($f_{\min}$) |
|---|---|---|---|
| $f_1(x) = \sum_{l=1}^{L} x_l^2$ | 10, 30 | $[-100, 100]^L$ | 0 |
| $f_2(x) = \sum_{l=1}^{L} \lvert x_l \rvert + \prod_{l=1}^{L} \lvert x_l \rvert$ | 10, 30 | $[-10, 10]^L$ | 0 |
| $f_3(x) = \sum_{l=1}^{L} (\sum_{g=1}^{l} x_g)^2$ | 10, 30 | $[-100, 100]^L$ | 0 |
| $f_4(x) = \sum_{l=1}^{L} (\lfloor x_l + 0.5 \rfloor)^2$ | 10, 30 | $[-100, 100]^L$ | 0 |
| $f_5(x) = \sum_{l=1}^{L} -x_l \sin\left(\sqrt{\lvert x_l \rvert}\right)$ | 10, 30 | $[-500, 500]^L$ | −12569.5 |
| $f_6(x) = -20 \exp\left(-0.2\sqrt{\left(\sum_{l=1}^{L} x_l^2\right)/L}\right)$<br>$- \exp\left((1/L) \sum_{l=1}^{L} \cos 2\pi x_l\right) + 20 + e$ | 10, 30 | $[-32, 32]^L$ | 0 |
| $f_7(x) = (1/4000) \sum_{l=1}^{L} x_l^2 - \prod_{l=1}^{L} \cos\left(x_l/\sqrt{l}\right) + 1$ | 10, 30 | $[-600, 600]^L$ | 0 |

represents that $\mathrm{DC}_1$ replenishes its item at every $2T$ interval; $\mathrm{DC}_2$ and $\mathrm{DC}_3$ replenish their item at each $T$ interval.

*The Initialization*. According to the representation of the decoded individual, the dimension of initial individual can be confirmed as $m + n + 3$. Then the values of $N_p$ individuals with $m + n + 3$ dimension are generated between 0 and 1 randomly and then are mapped into practical values (see Figure 3) through decoding scheme.

*3.2.2. Decoding Scheme.* The target of decoding is converting the initial chromosome to practical solution. Denote $x_{t,q}^G$ as the value of gene $q$ of individual $t$ in $G$ generation; $p_{t,q}^G$ as the practical value by decoding $x_{t,q}^G$; $p_q^L$ as the lower bound of $x_{\cdot,q}^G$; $p_q^U$ as the upper bound of $x_{\cdot,q}^G$. Since the practical values of the first two parts of chromosome are integer, the formulation of decoding is as follows:

$$p_{t,q}^G = \operatorname{round}\left(p_q^L + x_{t,q}^G \cdot \left(p_q^U - p_q^L\right)\right), \quad q = 1, \ldots, N_d, \qquad t = 1, \ldots, N_p;\; G = 1, \ldots GenM. \tag{14}$$

The practical values of the last two parts of chromosome are between 0 and 1, since the value of gene can directly map into practical value.

From (14), we can see that the lower and upper bound is the key for decoding. It is obvious that the lower bound of opened DC and basic order cycle time are equal to 1. As to the upper bound, if the value is set too small, the optimal solution may be excluded; if the value is set too big, the search scope is enlarged which has directly impact on performance of algorithm. The maximum number of opened DCs ($n$) is usually decided according to the number and scope of customers; the upper bound of replenishment frequency ($k_i$) is set to 15 which is almost 4 times of the maximum optimal replenishment frequency obtained by Silva and Gao [2].

## 4. Numerical Examples and Results Discussion

To verify the performance of the proposed HSDE, three numerical examples were designed. Another two intelligent algorithms, GAs and HDE which had been proven to be effective approaches for solving JRPs and JRDs [4, 19, 20], were chosen to compare with it. Example 1 is numerical optimization problems with benchmark functions; Example 2 is used to compare three algorithms by different sizes of the JR-LIPs; Example 3 is designed to observe the impact of different cost parameters on the optimal decision.

The decoding scheme of GA and HDE is the same with HSDE. Besides the number of populations ($N_p$) and maximum number of iterations (*GenM*) and the rest of common parameters for three numerical examples are set as follows: (a) for HSDE, the parameters had been given in Section 3.2.1; (b) for HDE, crossover rate CR is set to 0.3, $F$ is set to 0.6; (c) for GA, the probabilities of crossover and mutation are set to 0.9 and 0.1, respectively. The decisions for using these values are based on the experience from the works of literature [4, 29, 38].

### *4.1. Example 1: Comparative Study by Benchmark Functions*

*4.1.1. Seven Benchmark Functions.* In order to verify the performance of HSDE, seven well-known benchmark functions are used in the following experiments. To assure a relatively fair comparison, the functions were selected according to their different properties [29]. Functions $f_1$–$f_3$ are unimodal functions, $f_4$ is a step function, and $f_5$–$f_7$ are multimodal functions which appear to be the most difficult class of problems for many intelligent algorithms. The details about seven test functions are listed in Table 2 ($L$ denotes the dimension and $M$ denotes the decision space of the problem).

*4.1.2. Comparative Results and Analysis.* Three algorithms including HSDE, HDE, and GA are compared with two different dimensions, that is, the speed of convergence and the ability to obtain optimal solution. In all cases, the number of populations ($N_p$) is 100; the maximum number of iterations (*GenM*) is 300 and 500 for $L = 10$ and $L = 30$, respectively. The average $f_{\min}$± standard deviations of each algorithm for different test functions over 10 independent runs are shown in Tables 3 and 4. The best results are highlighted in bold face. Furthermore, in order to have an intuitive understanding about the convergence speed performance of each algorithm, Figures 4, 5, 6, and 7 present the average fitness curve of two unimodal functions ($f_1$ and $f_3$), step function $f_4$ and multimodal function $f_6$, with thirty dimensions.

Table 3: Results of seven functions with three algorithms ($L = 10$).

| Function | HSDE | HDE | GA |
|---|---|---|---|
| $f_1$ | **0** | **0** | 0.0714 ± 0.0341 |
| $f_2$ | **0** | **0** | 0.0482 ± 0.0090 |
| $f_3$ | **1.4868e − 06** ± 5.3600*e* − 06 | 8.3092*e* − 06 ± 2.6265*e* − 05 | 81.1709 ± 57.6638 |
| $f_4$ | **0** | **0** | 0.6928 ± 0.2275 |
| $f_5$ | −4.1646*e* + 03 ± 53.1791 | **−4.1898e + 03** ± 9.5869*e* − 13 | −3.7682*e* + 03 ± 0.3011 |
| $f_6$ | **8.8820e − 16 ± 2.0788e − 31** | **8.8820e − 16 ± 2.0788e − 31** | 0.1348 ± 0.0492 |
| $f_7$ | **0.0138** ± 0.0107 | 0.0143 ± 0.0084 | 0.1653 ± 0.0395 |

Table 4: Results of seven functions with three algorithms ($L = 30$).

| Function | HSDE | HDE | GA |
|---|---|---|---|
| $f_1$ | **1.4139e − 13** ± 1.2000*e* − 13 | 3.8990*e* − 12 ± 1.5278*e* − 12 | 133.5903 ± 37.2994 |
| $f_2$ | **8.3180e − 09** ± 4.2406*e* − 09 | 5.6406*e* − 08 ± 2.3898*e* − 08 | 2.7869 ± 0.4786 |
| $f_3$ | **5.1928e + 02** ± 1.1432*e* + 03 | 1.4403*e* + 03 ± 890.9681 | 3.8840*e* + 03 ± 1.7548*e* + 03 |
| $f_4$ | **0** | **0** | 148.8000 ± 43.3585 |
| $f_5$ | −1.1944*e* + 04 ± 439.6241 | **−1.2475e + 04** ± 93.0797 | −1.2417*e* + 04 ± 35.5129 |
| $f_6$ | **7.5180e − 08** ± 3.3095e − 08 | 3.9619*e* − 07 ± 1.1968*e* − 07 | 4.1132 ± 0.3369 |
| $f_7$ | **0.0012** ± 0.0032 | 0.0016 ± 0.0042 | 2.1764 ± 0.3635 |

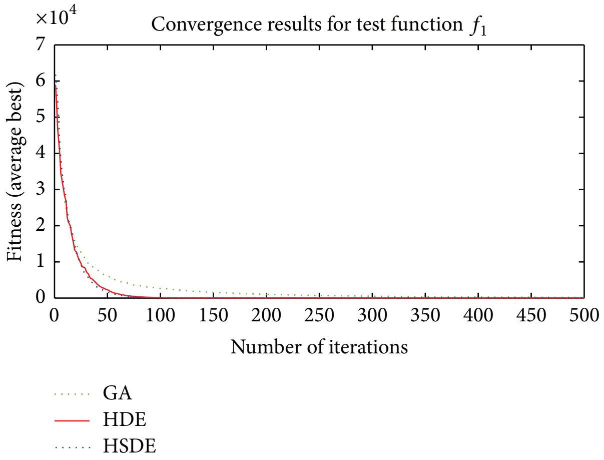


Figure 4: Convergence results of $f_1$ ($L = 30$).

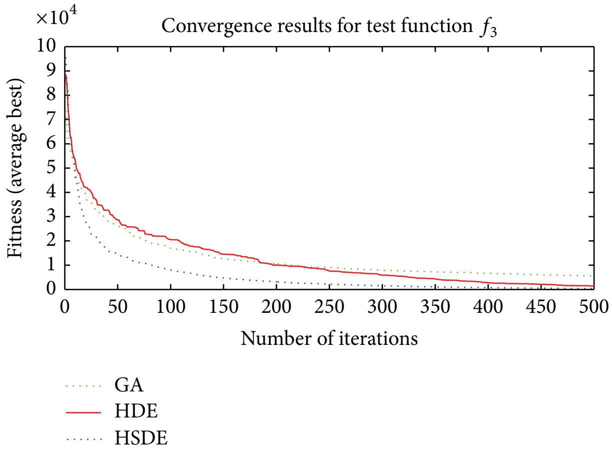


Figure 5: Convergence results of $f_3$ ($L = 30$).

Results in Tables 3 and 4 and Figures 4–7 show that

(1) the ability to obtain optimal solutions of HSDE is better than HDE and GA, especially for functions with the large dimension (when $L = 10$, the cases to obtain best performance of HSDE, HDE, and GA are 6, 5, 0; for $L = 30$, the cases are 6, 2, 0, resp.);

(2) although HSDE and HDE can both obtain the global optimum for $f_4$ with thirty dimensions, the convergence curves presented in Figure 6 illustrate that HSDE has better convergence speed. In summary, the proposed HSDE has better performance for the large-scale optimization problems.

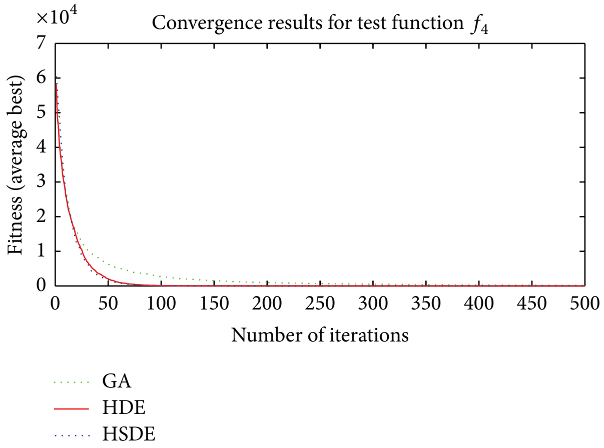


Figure 6: Convergence results of $f_4$ ($L = 30$).

Table 5: Parameters for test problems.

| Parameters | | Value |
|---|---|---|
| Common parameters | | |
| $D_j$, $j = 1, 2, \ldots, m$ | Annual demand rate | $U$ (80, 800) |
| $S$ | Major ordering cost | 45 |
| $s_l$, $l = 1, 2, \ldots, m$ | Minor ordering cost | $U$ (1, 10) |
| $h_l$, $l = 1, 2, \ldots, m$ | Annual inventory holding cost | $U$ (0, 1) |
| $f_l$, $l = 1, 2, \ldots, m$ | Fixed location cost | $U$ (400, 800) |
| Changed parameters | | |
| $n$ | Maximum number of opened DCs | 5, 10, 20 |
| $m$ | Number of customers (potential sites of DCs) | 30, 50, 100 |

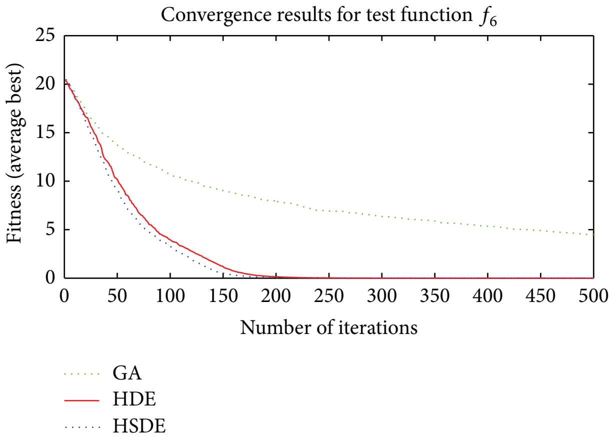


Figure 7: Convergence results of $f_6$ ($L = 30$).

*4.2. Example 2: Performance Comparison for the JR-LIPs*

*4.2.1. Basic Data for the JR-LIP.* In this section, three different scales of the JR-LIPs are used to test the performance of three intelligent algorithms. The part of basic data listed in Table 5 comes from Silva and Gao [2]. In all examples, the demand points were randomly located in a 50 × 50 square.

*4.2.2. Results and Analysis.* We denote $P\_n\_m$ as the scale of problems. As for $N_p$ of the HSDE and HDE, Storn and Price [38] proposed that the population size $N_p \in [4N_d, 10N_d]$; Nobakhti and Wang [39] and Wang et al. [40] suggested that $N_p \in [2N_d, 20N_d]$ is rational. According to their experiments, we set $N_p = 200$, 300, and 450 for $P\_5\_30$, $P\_10\_50$, and $P\_20\_100$, respectively. Table 6 reported the results of the average CPU times (Avg CPU times), the optimal total cost, the average minimum total cost for 20 times (Avg $\text{TC}_{\text{min}}$), and the ratio of finding the optimal total cost. Figures 8 and 9 showed the convergence trend of three algorithms (based on average fitness) for $P\_5\_30$ and $P\_10\_50$.

Results in Table 6 show that (1) both HSDE and HDE can find the optimal results with 100% for the smaller scale problem; (2) with the increasing of problem size, the performance of HSDE is better than HDE and GA whatever in CPU times and total cost. Furthermore, the running time of HSDE for $P\_10\_50$ (99 seconds) is superior to Silva and Gao [2] in which the running time is more than 270 seconds when $m = 50$.

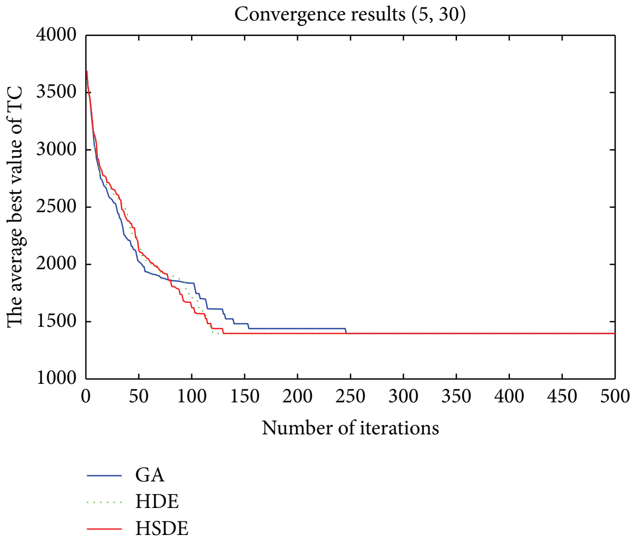


Figure 8: Convergence trend ($P\_5\_30$).

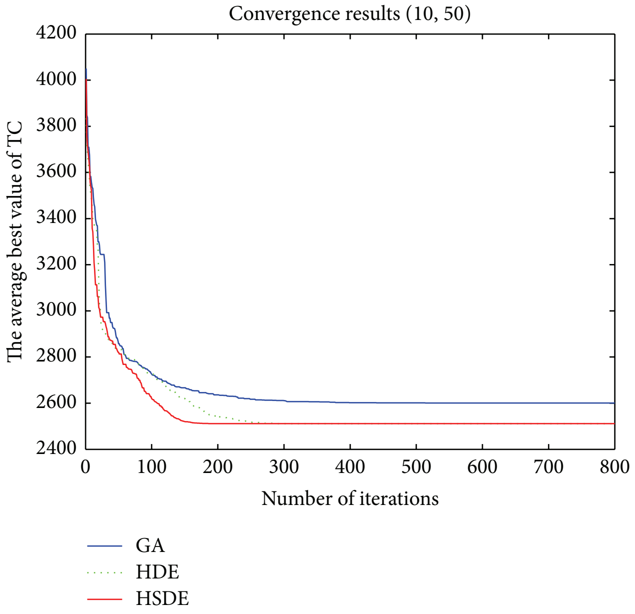


Figure 9: Convergence trend ($P\_10\_50$).

Table 6: Comparison of the results for different scales.

| *P*_*n*_*m* | Algorithm | Avg. CPU times | Optimal TC | Avg. $\mathrm{TC}_{\min}$ | Ratio of finding the optimal TC |
|---|---|---|---|---|---|
| *P*_5_30 | HSDE | 17.6 | 1397.4 | 1397.4 | 100% |
| | HDE | 17.3 | 1397.4 | 1397.4 | 100% |
| | GA | 19.2 | 1397.4 | 1397.4 | 100% |
| *P*_10_50 | HSDE | 99.0 | 2511.2 | 2511.2 | 100% |
| | HDE | 105.1 | 2511.2 | 2511.2 | 100% |
| | GA | 114.1 | 3064.8 | 3126.0 | 0% |
| *P*_20_100 | HSDE | 613.2 | 4444.7 | 4709.2 | 45% |
| | HDE | 689.9 | 7318.3 | 7846.4 | 0% |
| | GA | 742.3 | 8539.4 | 9497.5 | 0% |

Table 7: Comparative results using three algorithms under different $h_l$.

| $\Delta h_l$ (%) | Algorithms | Location sites (DCs) | CPU time | $k_i$ | $C_L$ | $C_R$ | $C_L + C_R$ |
|---|---|---|---|---|---|---|---|
| −40 | HSDE | 36, 39, 32 | 85.6 | 1, 1, 1 | 2174.7 | 298.7 | 2473.4 |
| | HDE | 32, 39, 36 | 81.0 | 1, 1, 1 | 2174.7 | 298.7 | 2473.4 |
| −20 | HSDE | 32, 39, 36 | 89.5 | 1, 1, 1 | 2177.0 | 342.3 | 2519.3 |
| | HDE | 32, 39, 36 | 88.3 | 1, 1, 1 | 2177.0 | 342.3 | 2519.3 |
| 20 | HSDE | 39, 36, 32 | 106.4 | 1, 1, 1 | 2177.0 | 419.2 | 2596.2 |
| | HDE | 32, 39, 36 | 113.6 | 1, 1, 1 | 2177.0 | 419.2 | 2596.2 |

Table 8: Comparative results using three algorithms under different *S*.

| $\Delta S$ (%) | Algorithms | Location sites (DCs) | CPU time | $k_i$ | $C_L$ | $C_R$ | $C_L + C_R$ |
|---|---|---|---|---|---|---|---|
| −40 | HSDE | 31, 9, 3 | 105.4 | 2, 1, 1 | 2888.9 | 394.5 | 3283.4 |
| | HDE | 3, 31, 9 | 109.9 | 2, 2, 1 | 2881.4 | 403.0 | 3284.4 |
| −20 | HSDE | 49, 31, 3 | 96.7 | 1, 1, 1 | 2951.2 | 358.2 | 3309.4 |
| | HDE | 3, 31, 9 | 106.8 | 1, 1, 1 | 2924.2 | 401.2 | 3325.4 |
| 20 | HSDE | 31, 49, 3 | 107.9 | 1, 1, 1 | 2966.1 | 429.2 | 3395.3 |
| | HDE | 31, 49, 3 | 111.5 | 1, 1, 1 | 2966.1 | 429.2 | 3395.3 |

Figures 8 and 9 also show that HSDE is more stable and suitable for the large-scale problem.

Section 4.2 mainly focuses on the performance of algorithm in handling different scales of JR-LIPs. The more detailed analysis about the impacts of cost parameters on the optimal decision is presented in Section 4.3.

*4.3. Example 3: Impacts of Cost Parameters on Optimal Decision.* The size of problem in this section is *P*_10_50 which is the middle scale of three problems in Section 4.2 and the optimal TC can be found at each running by HSDE and HDE. GA is not used in this section due to its inferior performance (the ratio to find optimal TC is 0%). Tables 7 and 8 show the computational results by varying the values of inventory holding cost and major ordering cost. From the tables we can see clearly the impacts of cost parameters on DC selection, replenishment frequency ($k_i$), location costs ($C_L$), replenishment costs ($C_R$), and total running time of algorithms.

The comparative results of Tables 7 and 8 show the following useful conclusions.

(1) The robustness of HSDE is better than HDE.

(2) When inventory holding cost and major ordering cost vary from −40% to 20%, the optimal number of DCs is always three. Moreover, the location sites of DCs have little change ($\Delta S = 20\%$), and the varied direction of replenishment cost is the same with the cost parameters.

(3) DCs have more motivation to order item jointly when the major ordering cost S increases ($k_1 = k_2 = k_3 = 1$ indicates that DCs replenish jointly in each order). This observation is consistent with the principle of JR policy.

## 5. Conclusions and Future Research

In this paper, we proposed an effective intelligent algorithm for a modified joint replenishment and location-inventory (JR-LIP) model. The objective of the JR-LIP is to determine the number and locations of DCs, the assignment decision, and replenishment policy to minimize the total system cost. To handle this NP-hard problem effectively, an intelligent algorithm named HSDE is designed to solve the proposed model. To verify the effectiveness of HSDE, GA and HDE were chosen to be compared with it by benchmark functions tests and numerical examples. We can easily come to useful conclusions and managerial insight as follows.

(1) Results of benchmark functions tests show the good ability of HSDE in handling the large-scale problems. When the dimension of test function is 10, HSDE and HDE have the same precision, while the dimension is 30, HSDE has the higher precision and faster speed than HDE.

(2) The similar conclusion can be obtained from example 2. The rate of convergence obtained by HSDE and HDE is both 100% when $m = 30$ and $m = 50$, whereas when $m = 100$, the rate of HDE is 0% and HSDE is 45%.

(3) Example 3 illustrates the impacts of cost parameters on the optimal decision and reveals that when the major ordering cost is bigger, DCs have more incentive to replenish jointly for sharing related costs.

All numerical examples verify that the HSDE is an easy and effective algorithm to handle the JR-LIP. To our best knowledge, this is the first time to use the improved DE-based algorithm to solve this NP-hard problem.

In our model, the demand is constant which generated from a uniform distribution, and there is no capacity limitation. These assumptions are not required. The future direction about this problem includes relaxing the assumption to match real-world scenario and looking for more quickly and efficiently solving method.

## Acknowledgments

The authors are very grateful for the constructive comments of editors and referees. This research is partially supported by the National Natural Science Foundation of China (70801030; 71131004; 71371080; 71373093), Humanities and Social Sciences Foundation of Chinese Ministry of Education (No. 11YJC630275), and Fundamental Research Funds for the Central Universities (HUST: 2012TS065).

## References

[1] M. Khouja and S. Goyal, "A review of the joint replenishment problem literature: 1989–2005," *European Journal of Operational Research*, vol. 186, no. 1, pp. 1–16, 2008.

[2] F. Silva and L. Gao, "A joint replenishment inventory-location model," *Networks and Spatial Economics*, vol. 13, no. 1, pp. 107–122, 2013.

[3] J. Shu, C.-P. Teo, and Z.-J. M. Shen, "Stochastic transportation-inventory network design problem," *Operations Research*, vol. 53, no. 1, pp. 48–60, 2005.

[4] L. Wang, C. X. Dun, W. J. Bi, and Y. R. Zeng, "An effective and efficient differential evolution algorithm for the integrated stochastic joint replenishment and delivery model," *Knowledge-Based Systems*, vol. 36, pp. 104–114, 2012.

[5] W. W. Qu, J. H. Bookbinder, and P. Iyogun, "Integrated inventory-transportation system with modified periodic policy for multiple products," *European Journal of Operational Research*, vol. 115, no. 2, pp. 254–269, 1999.

[6] B. C. Cha, I. K. Moon, and J. H. Park, "The joint replenishment and delivery scheduling of the one-warehouse, n-retailer system," *Transportation Research E*, vol. 44, no. 5, pp. 720–730, 2008.

[7] I. K. Moon, B. C. Cha, and C. U. Lee, "The joint replenishment and freight consolidation of a warehouse in a supply chain," *International Journal of Production Economics*, vol. 133, no. 1, pp. 344–350, 2011.

[8] O. Berman, D. Krass, and M. M. Tajbakhsh, "A coordinated location-inventory model," *European Journal of Operational Research*, vol. 217, no. 3, pp. 500–508, 2012.

[9] C. P. Teo, J. Ou, and M. Goh, "Impact on inventory costs with consolidation of distribution centers," *IIE Transactions*, vol. 33, no. 2, pp. 99–110, 2001.

[10] M. S. Daskin, C. R. Coullard, and Z.-J. M. Shen, "An inventory-location model: formulation, solution algorithm and computational results," *Annals of Operations Research*, vol. 110, no. 1–4, pp. 83–106, 2002.

[11] Z.-J. M. Shen, C. Coullard, and M. S. Daskin, "A joint location-inventory model," *Transportation Science*, vol. 37, no. 1, pp. 40–55, 2003.

[12] Z.-J. Shen and L. Qi, "Incorporating inventory and routing costs in strategic location models," *European Journal of Operational Research*, vol. 179, no. 2, pp. 372–389, 2007.

[13] L. V. Snyder, M. S. Daskin, and C.-P. Teo, "The stochastic location model with risk pooling," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1221–1238, 2007.

[14] L. Ozsen, C. R. Coullard, and M. S. Daskin, "Capacitated warehouse location model with risk pooling," *Naval Research Logistics*, vol. 55, no. 4, pp. 295–312, 2008.

[15] E. Arkin, D. Joneja, and R. Roundy, "Computational complexity of uncapacitated multi-echelon production planning problems," *Operations Research Letters*, vol. 8, no. 2, pp. 61–66, 1989.

[16] J. Krarup and P. M. Pruzan, "The simple plant location problem: survey and synthesis," *European Journal of Operational Research*, vol. 12, no. 1, pp. 36–81, 1983.

[17] A. Atamtürk, G. Berenguer, and Z. J. M. Shen, "A conic integer programming approach to stochastic joint location-inventory problems," *Operations Research*, vol. 60, no. 2, pp. 366–381, 2012.

[18] M. Shahabi, S. Akbarinasaji, A. Unnikrishnan, and R. James, "Integrated inventory control and facility location decisions in a multi-echelon supply chain network with hubs," *Networks and Spatial Economics*, 2013.

[19] S.-P. Hong and Y.-H. Kim, "A genetic algorithm for joint replenishment based on the exact inventory cost," *Computers and Operations Research*, vol. 36, no. 1, pp. 167–175, 2009.

[20] T. F. Abdelmaguid and M. M. Dessouky, "A genetic algorithm approach to the integrated inventory-distribution problem," *International Journal of Production Research*, vol. 44, no. 21, pp. 4445–4464, 2006.

[21] L. Wang, Q. L. Fu, C. G. Lee, and Y. R. Zeng, "Model and algorithm of fuzzy joint replenishment problem under credibility measure on fuzzy goal," *Knowledge-Based Systems*, vol. 39, pp. 57–66, 2013.

[22] L. Wang, C. X. Dun, C. G. Lee, Q. L. Fu, and Y. R. Zeng, "Model and algorithm for fuzzy joint replenishment and delivery scheduling without explicit membership function," *International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9–12, pp. 1907–1920, 2013.

[23] W. Qian and A. li, "Adaptive differential evolution algorithm for multiobjective optimization problems," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 431–440, 2008.

[24] Q.-K. Pan, L. Wang, and B. Qian, "A novel differential evolution algorithm for bi-criteria no-wait flow shop scheduling problems," *Computers and Operations Research*, vol. 36, no. 8, pp. 2498–2511, 2009.

[25] L. Wang, H. Qu, Y. H. Li, and J. He, "Modeling and optimization of stochastic joint replenishment and delivery scheduling problem with uncertain costs," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 657465, 12 pages, 2013.

[26] H. Qu, L. Wang, and Y.-R. Zeng, "Modeling and optimization for the joint replenishment and delivery problem with heterogeneous items," *Knowledge-Based Systems*, pp. 207–215, 2013.

[27] L. Wang, Q.-L. Fu, and Y.-R. Zeng, "Continuous review inventory models with a mixture of backorders and lost sales under fuzzy demand and different decision situations," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4181–4189, 2012.

[28] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.

[29] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[30] D. He, F. Wang, and Z. Mao, "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect," *International Journal of Electrical Power and Energy Systems*, vol. 30, no. 1, pp. 31–38, 2008.

[31] W.-Y. Lin, "A GA-DE hybrid evolutionary algorithm for path synthesis of four-bar linkage," *Mechanism and Machine Theory*, vol. 45, no. 8, pp. 1096–1107, 2010.

[32] S. Haijun and Y. Yupu, "Quantum-inspired differential evolution for binary optimization," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, vol. 1, pp. 341–346, October 2008.

[33] Q. Xu and J. Guo, "A quantum differential evolution algorithm for function optimization," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10)*, pp. V8-347–V8-350, October 2010.

[34] A. Pat, A. R. Hota, and A. Singh, "Quantum-inspired differential evolution on bloch coordinates of qubits," *Communications in Computer and Information Science*, vol. 125, pp. 18–24, 2011.

[35] T. Zheng and M. Yamashiro, "Solving flow shop scheduling problems by quantum differential evolutionary algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5-8, pp. 643–662, 2010.

[36] G. Zhang, "Quantum-inspired evolutionary algorithms: a survey and empirical study," *Journal of Heuristics*, vol. 17, no. 3, pp. 303–351, 2011.

[37] P. Luukka, "Similarity classifier using similarities based on modified probabilistic equivalence relations," *Knowledge-Based Systems*, vol. 22, no. 1, pp. 57–62, 2009.

[38] 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.

[39] A. Nobakhti and H. Wang, "A simple self-adaptive Differential Evolution algorithm with application on the ALSTOM gasifier," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 350–370, 2008.

[40] L. Wang, J. He, and Y.-R. Zeng, "A differential evolution algorithm for joint replenishment problem using direct grouping and its application," *Expert Systems*, vol. 29, no. 5, pp. 429–441, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 378515, 11 pages
http://dx.doi.org/10.1155/2013/378515

*Research Article*

# Seven-Spot Ladybird Optimization: A Novel and Efficient Metaheuristic Algorithm for Numerical Optimization

**Peng Wang, Zhouquan Zhu, and Shuai Huang**

*School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Peng Wang; wangpeng305@nwpu.edu.cn

Received 9 September 2013; Accepted 17 October 2013

Academic Editors: T. Chen and J. Yang

This paper presents a novel biologically inspired metaheuristic algorithm called seven-spot ladybird optimization (SLO). The SLO is inspired by recent discoveries on the foraging behavior of a seven-spot ladybird. In this paper, the performance of the SLO is compared with that of the genetic algorithm, particle swarm optimization, and artificial bee colony algorithms by using five numerical benchmark functions with multimodality. The results show that SLO has the ability to find the best solution with a comparatively small population size and is suitable for solving optimization problems with lower dimensions.

## 1. Introduction

In recent years, heuristic algorithms have gained popularity because of their ability to find near-optimal solutions to problems unsolved by analytical methods within reasonable computation time due to the multimodality or high dimensionality of their objective functions [1]. Heuristic algorithms are usually developed to solve a specific problem. There is also a class of heuristic algorithms which can be used to solve a large class of problems either directly or with minor modifications, hence the name meta-heuristic algorithms [2].

Researchers continue to develop many meta-heuristic algorithms. Some of the most successful meta-heuristic algorithms include genetic algorithm (GA) [3], ant colony optimization [4], particle swarm optimization (PSO) [5], and artificial bee colony (ABC) [6]. Some of the recently proposed meta-heuristic algorithms include cuckoo search [7], monkey search [8], firefly algorithm [9], grenade explosion method [10], cat swarm optimization [11], and the artificial chemical reaction optimization algorithm [12]. The majority of these algorithms are biologically inspired; that is, they mimic nature for problem solving.

Meta-heuristic algorithms are widely used in different fields and problems, such as manufacturing [13], services [14], scheduling [15], transportation [16], health [17], sports [18], geology [19], and astronomy [20]. A single meta-heuristic algorithm that can solve all optimization problems of different types and structures does not exist, and, thus, new meta-heuristic optimization algorithms are continuously developed [21].

This paper introduces a novel biologically inspired meta-heuristic algorithm called seven-spot ladybird optimization (SLO). SLO is inspired by the foraging behavior of a seven-spot ladybird. This paper presents the basic concepts and main steps of the SLO and demonstrates its efficiency. The performance of the SLO is compared with some popular meta-heuristic algorithms, such as GA, PSO, and ABC, by using five different dimensional classical benchmark functions, as given in [6, 22]. The simulated results show that the SLO has the ability to get out of a local minimum and is efficient for some multivariable, multimodal function optimizations.

In general, all the metaheuristic algorithms have something in common in the sense that they are population-based search methods. These methods move from a set of points (population) to another set of points in a single iteration with likely improvement using a combination of deterministic or probabilistic rules. The most remarkable difference of these metaheuristic algorithms lies in the updating rules. The GA is inspired by the principles of genetics and evolution and

Figure 1: Seven-spot ladybird.

mimics the reproduction behavior observed in biological populations. The GA employs the principal of "survival of the fittest" in its search process to select and generate individuals that are adapted to their environment. In PSO, instead of using genetic operators, each particle adjusts its "flying" according to its own flying experience and its companions' flying experience [23]. ABC uses minimal model that mimics the foraging behavior of bees comprising of employed bees, onlooker bees and scouts [24]. The bees aim at discovering places of food sources with high amount of nectar (good fitness values). Differently, in our paper, SLO attempts to simulate the foraging behavior of a seven-spot ladybird which is rarely researched in the field of metaheuristic algorithm. The SLO algorithm consists of three essential components: dividing patches, searching food, and dispersal. Dividing patches increases the search efficiency and dispersal progressively reduces the search space. The search strategy in our algorithm is classified into extensive search and intensive search. Extensive search overcomes the weakness of local search and intensive search increases the possibility of achieving latent best solution. All the ideas are inspired by recent discoveries on the foraging behavior of a seven-spot ladybird which are quite different from other metaheuristic algorithms.

The rest of this paper is organized as follows. Section 2 presents the foraging behavior of the seven-spot ladybird. Section 3 describes the SLO and the steps in detail. Section 4 discusses the experiments and the results. Section 5 draws the conclusions.

## 2. Seven-Spot Ladybird Foraging Behaviors

The seven-spot ladybird (Figure 1), *Coccinella septempunctata*, is a common, easily recognizable insect that has attracted a considerable amount of interest from professional entomologists.

Recent studies have shown that seven-spot ladybirds are more social than we believe them to be [25–28]. Seven-spot ladybirds use different kinds of *pheromones* at different stages of its life history, such as eggs, larvae, pupa, and adult stages (Figure 2). Some chemical ecologies of the seven-spot ladybirds, with special attention to semiochemicals involved in *social communication* and *foraging behaviors*, have been reviewed in [29].

Seven-spot ladybirds are effective predators of aphids and other homopteran pests, and, thus, their foraging behaviors have been extensively studied [30–35]. Some scholars classified the *environmental levels* of seven-spot ladybirds into prey, patches, and habitats (Figure 3) [33–35], providing a framework for discussing the foraging behaviors of seven-spot ladybirds.

In Figure 3, movement between prey within aggregates of aphids is referred to as *intensive search* which is slow and sinuous. Movement between aggregates within a patch is referred to as *extensive search* which is relatively linear and fast. Movement between patches is called *dispersal* and movement from patches to hibernation is called *migration*.

Seven-spot ladybirds locate their prey via extensive search and then switch to intensive search after feeding. While searching for its prey, a seven-spot ladybird holds its antennae parallel to its searching substratum and its maxillary palpi perpendicular to the substratum. The ladybird vibrates its maxillary palpi and turns its head from side to side. The *sideward vibration* can increase the area wherein the prey may be located.

How seven-spot ladybirds decide when to leave a patch for another, also known as *dispersal*, remains unclear. Several authors suggested that beetles decide to leave when the capture rate falls below a critical value or when the time since the last aphid was captured exceeds a certain threshold [36–38].

## 3. Proposed Seven-Spot Ladybird Optimization (SLO) Algorithm

This section describes the proposed seven-spot ladybird optimization (SLO) algorithm, which simulates the foraging behavior of seven-spot ladybirds to solve multidimensional and multimodal optimization problems. The main steps of the SLO are as follows.

*Step 1* (dividing patches). Suppose that the search space (environment) is a $D$-dimensional space. The $i$th dimensional space is divided into $n_i$ subspaces, and the whole dimensional space is divided into $n = \Pi n_i$ subspaces (patches).

*Step 2* (initializing population). Suppose that each seven-spot ladybird is treated as a point in a $D$-dimensional patch. The $i$th ladybird is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where $X_i$ is a latent solution to the optimized question.

If $m$ is the number of seven-spot ladybirds initialized with random positions in a patch, then the population size of the seven-spot ladybirds is $N$, $N = m \times n$.

*Step 3* (calculating fitness). For each particle, evaluate the optimization fitness in a $D$-dimensional patch.

*Step 4* (choosing the best ladybird). The current fitness evaluation of each ladybird was compared with the fitness value of its best historical position ($s$best). If the current value is better than the previous one, then set $s$best value is equal to

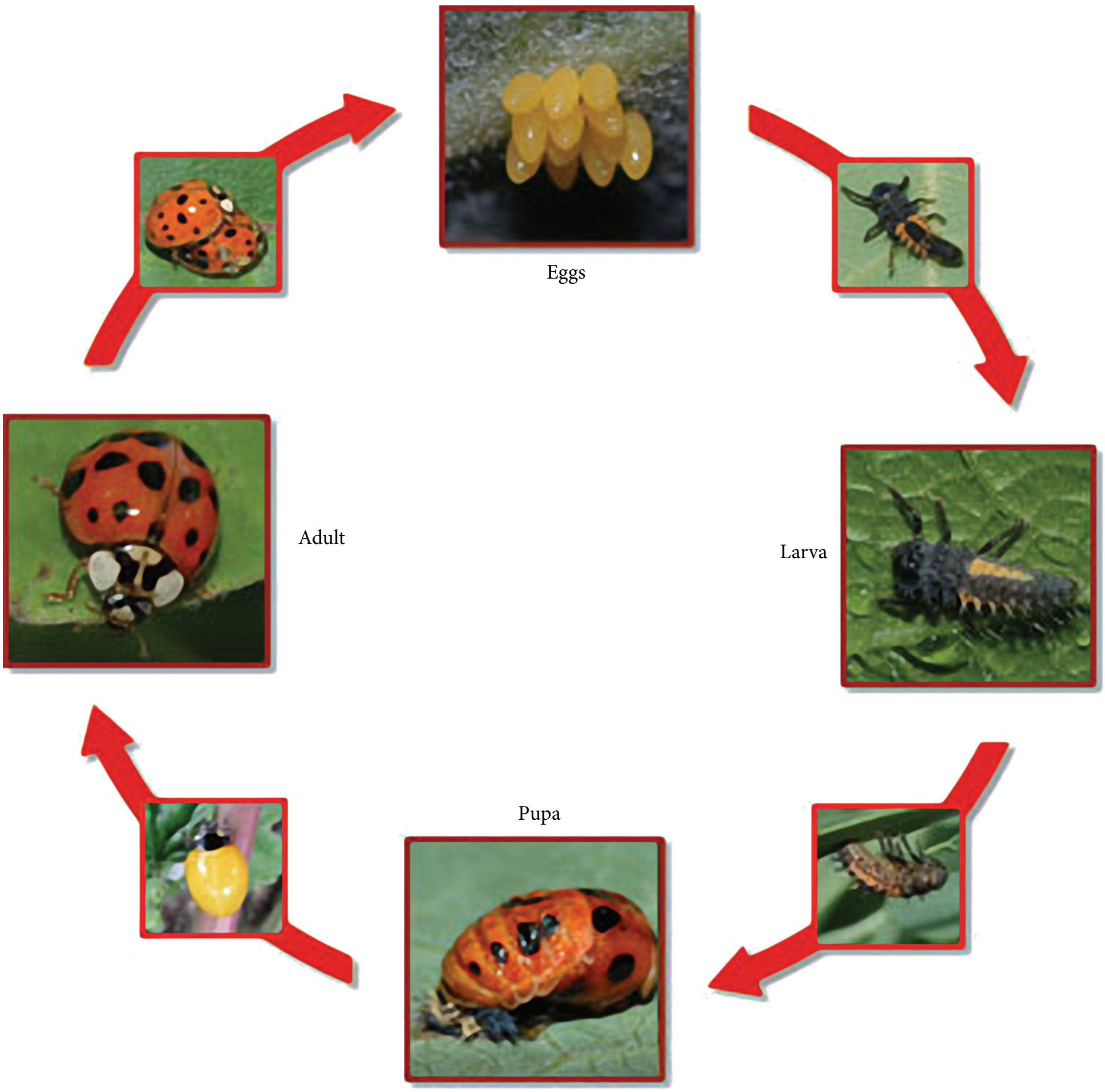


FIGURE 2: The life history of the seven-spot ladybird.

the current value, and the *s*best position is equal to the current position.

The current best fitness evaluation of all the ladybirds in a patch was compared with the fitness value of their previous best position (*l*best). If current value is better than the previous one, then set *l*best value equal to the current value, and the *l*best position equal to the current position.

The current best fitness evaluation of all the ladybirds in the population was compared with the fitness value of their previous best position (*g*best). If the current value is better than the previous one, then set *g*best value equal to the current value, and the *g*best position equal to the current position.

*Step 5* (dispersal). In the SLO, if a position does not improve in a predetermined number of cycles, then a new position is produced in the patch where *g*best exists, replacing the abandoned position. The new position is produced near the *g*best to share the information of the best ladybird in the whole particle. The value of the predetermined number of cycles (*limit*) is an important control parameter in the SLO.

If the abandoned position is $X_i$ and $j \in \{1, 2, \ldots, D\}$, then the seven-spot ladybird discovers a new position $X_i'$ as follows:

$$x_{i,j}' = x_{g\text{best},j} + \phi w, \tag{1}$$

where $w$ is the neighborhood space of *g*best and $\phi$ is a random number between $[-1, 1]$.

*Step 6* (updating positions). The position of a ladybird is updated associated with its previous movement. If a ladybird has done extensive search, then the position of the ladybird is changed as follows:

$$V_i(t) = c * r_1 * (S_i(t) - X_i(t)) + \varepsilon_1, \tag{2}$$

$$X_i(t+1) = X_i(t) + V_i(t), \qquad |V_i(t)| \le V_{\max}. \tag{3}$$

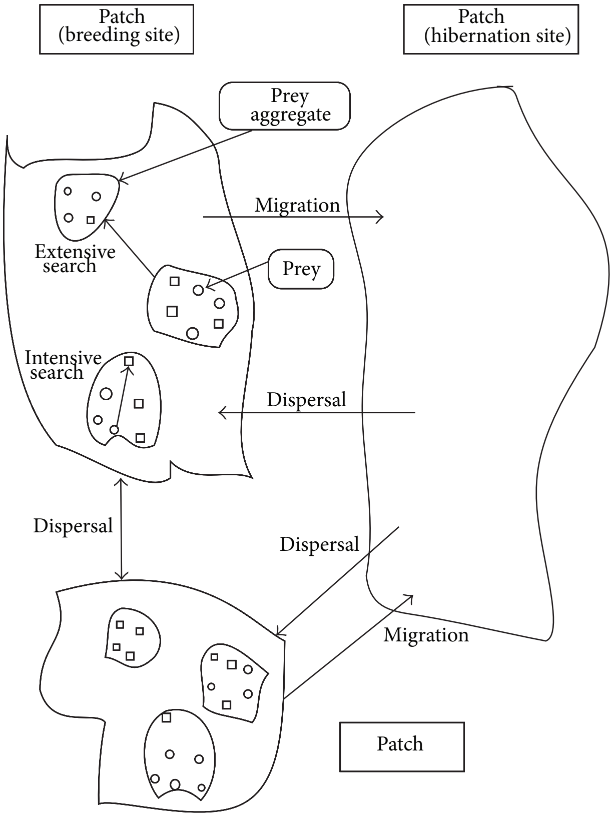


FIGURE 3: Diagram illustrating how a ladybird might perceive its environment and forage for resources.

After intensive search, a ladybird switches to extensive search. The position is updated according to the following equations:

$$V_i(t) = c * r_2 * (L_i(t) - X_i(t)) + \varepsilon_2, \quad (4)$$

$$X_i(t+1) = X_i(t) + V_i(t), \qquad |V_i(t)| \le V_{\max}. \quad (5)$$

In (2) and (4), $r_1$ and $r_2$ are two random numbers uniformly distributed from 0 to 1 and the positive constant $c$ is used for adjusting the search step and search direction in each iteration. In (3) and (5), the velocities of the ladybirds in each dimension are limited to the maximum velocity $V_{\max}$, which decides the search precision of the ladybirds in a solution space. If $V_{\max}$ is too high, then the ladybirds will possibly fly over the optimal solution. However, if the $V_{\max}$ is too low, then the ladybirds will fall into the local search space and have no method to carry on with the global search. Typically, $V_{\max}$ is set as follows:

$$V_{\max} = 0.2\,(\text{ub} - \text{lb}), \quad (6)$$

where ub and lb are the upper and lower bounds of each patch, respectively. Equation (6) came from [39]. We adopt it here to clamp the particles' velocities on each dimension.

From equations above, we can see that the velocity updating rule is composed of three parts. The first part, known as *intensive search*, is inspired by the slow and sinuous movements of ladybirds. The second part, known as *extensive search*, is derived from the relatively linear and fast movement behavior of ladybirds. The third part imitates the *sideward vibration* of ladybirds to increase the search area where the potential solution may exist. The parameter $\varepsilon_1$ and $\varepsilon_2$ are usually set as relatively small random numbers.

*Step 7* (inspecting termination condition). If the termination condition is satisfied, that is, the SLO has achieved the maximum iteration number, then the SLO is terminated; otherwise, it returns to Step 3.

## 4. Experiments

*4.1. Benchmark Functions.* In the field of heuristic computation, it is common to compare different algorithms using a set of test functions. However, the effectiveness of an algorithm against another algorithm cannot be measured by the number of problems that it solves better [40]. In this way, we have made a previous study of the functions to be optimized for constructing a test set with fewer functions and a better selection. We used five classical benchmark functions to compare the performance of the proposed SLO with those of GA, PSO, and ABC. This set is adequate to include different kinds of problems such as unimodal, multimodal, regular, irregular, separable, nonseparable, and multidimensional. Mathematical descriptions of the benchmark functions were obtained from [6, 22].

The first function is the Griewank function whose value is 0 at its global minimum $(0, 0, \ldots, 0)$ (7). Initialization range for the function is $[-600, 600]$. The Griewank function has a product term that introduces interdependence among its variables. The aim is to overcome the failure of the techniques that optimize each variable independently. The optima of the Griewank function are regularly distributed. Since the number of local optima increases with the dimensionality, this function is strongly multimodal. The multimodality disappears for sufficiently high dimensionalities ($n > 30$) and makes the problem unimodal. Consider

$$f_1(\vec{x}) = \frac{1}{4000}\left(\sum_{t=1}^{D}\left(x_i^2\right)\right) - \left(\prod_{t=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)\right) + 1. \quad (7)$$

The second function is the Rastrigin function whose value is 0 at its global minimum $(0, 0, \ldots, 0)$ (8). Initialization range for the function is $[-15, 15]$. The Rastrigin function is based on the Sphere function with the addition of cosine modulation to produce many local minima, making it multimodal. The locations of the minima are regularly distributed. The difficult part about finding optimal solutions to the Rastrigin function is that an optimization algorithm is easily trapped in a local optimum on its way towards the global optimum. Consider

$$f_2(\vec{x}) = \sum_{t=1}^{D}\left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right). \quad (8)$$

The third function is the Rosenbrock function whose value is 0 at its global minimum $(1, 1, \ldots, 1)$ (9). Initialization range for the function is [−15, 15]. The global optimum is inside a long, narrow, parabolic-shaped flat valley. Since it is difficult to converge to the global optimum, the variables are strongly dependent, and the gradients generally do not point towards the optimum, this problem is repeatedly used to test the performance of the optimization algorithms. Consider

$$f_3(\vec{x}) = \sum_{t=1}^{D} 100\left(x_i^2 - x_{i+1}\right)^2 + \left(1 - x_i\right)^2. \tag{9}$$

The fourth function is the Ackley function whose value is 0 at its global minimum (0, 0, . . . , 0) (10). Initialization range for the function is [−32.768, 32.768]. The Ackley function has an exponential term that covers its surface with numerous local minima, making its complexity moderated. An algorithm that only uses the gradient steepest descent will be trapped in the local optima, but any search strategy that analyzes a wider region will be able to cross the valley among the optima and achieve better results. A search strategy must combine the exploratory and exploitative components efficiently to obtain good results for the Ackley function. Consider

$$\begin{aligned} f_4(\vec{x}) &= 20 + e - 20e^{\left(-0.2\sqrt{(1/D)\sum_{t=1}^{D} x_i^2}\right)} \\ &\quad - e^{(1/D)\sum_{t=1}^{D}\cos(2\pi x_i)}. \end{aligned} \tag{10}$$

The fifth function is the Schwefel function whose value is 0 at its global minimum (420.9867, 420.9867, . . . , 420.9867) (11). Initialization range for the function is [−500, 500]. The surface of the Schwefel function is composed of a large number of peaks and valleys. The Schwefel function has a second best minimum far from the global minimum where many search algorithms are trapped. Moreover, the global minimum is near the bounds of the domain. Consider

$$f_5(\vec{x}) = D * 418.9829 + \sum_{t=1}^{D} - x_i \sin\left(\sqrt{|x_i|}\right). \tag{11}$$

*4.2. Settings for Algorithms.* The common control parameters for the algorithms include population size and number of maximum generation. In the experiments, maximum generations were 750, 1000, and 1500 for Dimensions 5, 10, and 30, respectively, and the population size was 50. Other control parameters of the algorithms and the schemes used in [6], including the control parameter values employed for GA, PSO, and ABC are presented below.

*4.2.1. GA Settings.* The settings for the used GA scheme presented in [6] are as follows: single point uniform crossover with rate of 0.95, random selection mechanism, Gaussian mutation with rate of 0.1, and linear ranking fitness function. A child chromosome is added to the population by using the child production scheme.

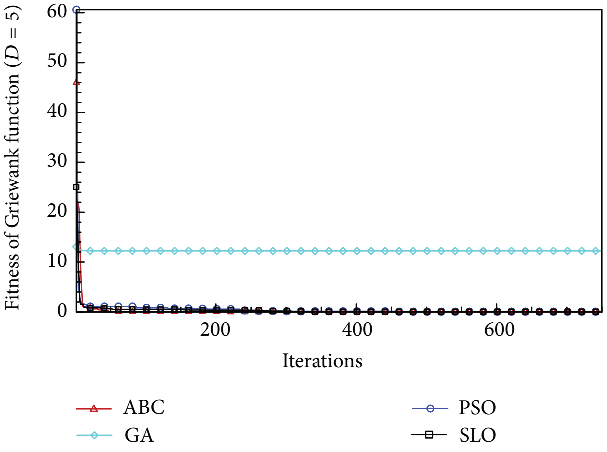


FIGURE 4: Convergence characteristics of the Griewank function with $D = 5$.

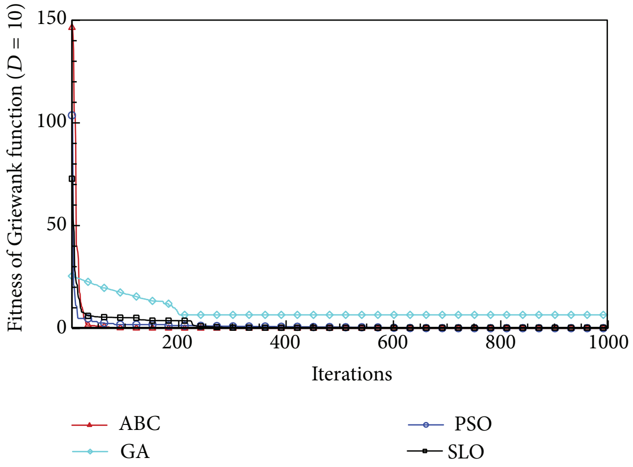


FIGURE 5: Convergence characteristics of the Griewank function with $D = 10$.

*4.2.2. PSO Settings.* PSO equations can be expressed as follows:

$$\begin{aligned} \vec{v}(t+1) &= w\vec{v}(t) + \phi_1 \text{rand}(0,1)\left(\vec{p}(t) - \vec{x}(t)\right) \\ &\quad + \phi_2 \text{rand}(0,1)\left(\vec{g}(t) - \vec{x}(t)\right), \\ \vec{x}(t+1) &= \vec{x}(t) + \vec{v}(t+1), \end{aligned} \tag{12}$$

where $w$ is the additional inertia weight that varies from 0.9 to 0.4 linearly with the iterations. The learning factors, $\phi_1$ and $\phi_2$, are set to 2. The upper and lower bounds for $v$, $(v_{\min}, v_{\max})$ are set as the maximum upper and lower bounds of $x$; that is, $(v_{\min}, v_{\max}) = (x_{\min}, x_{\max})$. If the sum of accelerations would cause the velocity on that dimension $v(t+1)$ to exceed $v_{\min}$ or $v_{\max}$, then the velocity on that dimension $v(t+1)$ will be limited to $v_{\min}$ or $v_{\max}$, respectively [6].

Table 1: Results of the Griewank function.

| Algorithm | Dimension | Mean | Best | SD |
|---|---|---|---|---|
| SLO | 5 | $1.2870E - 01$ | $7.4000E - 03$ | $9.1200E - 02$ |
| | 10 | $3.1000E - 01$ | $2.4600E - 02$ | $5.2890E - 01$ |
| | 30 | $1.4705E + 00$ | $6.2000E - 03$ | $1.7894E + 00$ |
| GA | 5 | $1.2240E + 01$ | $1.2240E + 01$ | $1.2267E - 10$ |
| | 10 | $1.4568E + 01$ | $6.5624E + 00$ | $4.7149E + 00$ |
| | 30 | $1.4100E - 02$ | $1.9119E - 10$ | $2.4700E - 02$ |
| PSO | 5 | $2.3700E - 02$ | $7.4000E - 03$ | $1.2400E - 02$ |
| | 10 | $7.6900E - 02$ | $2.7000E - 02$ | $3.3600E - 02$ |
| | 30 | $1.3200E - 02$ | $1.1102E - 15$ | $1.4900E - 02$ |
| ABC | 5 | $6.8808E - 04$ | $0.0000E + 00$ | $2.1000E - 03$ |
| | 10 | $2.8000E - 03$ | $1.1102E - 16$ | $4.6000E - 03$ |
| | 30 | $6.3515E - 04$ | $0.0000E + 00$ | $3.5000E - 03$ |

Table 2: Results of the Rastrigin function.

| Algorithm | Dimension | Mean | Best | SD |
|---|---|---|---|---|
| SLO | 5 | $0.0000E + 00$ | $0.0000E + 00$ | $0.0000E + 00$ |
| | 10 | $2.2781E + 01$ | $9.9760E - 01$ | $1.4721E + 01$ |
| | 30 | $3.6123E + 02$ | $2.3147E + 02$ | $5.9041E + 01$ |
| GA | 5 | $6.3010E - 01$ | $2.0653E - 09$ | $6.1180E - 01$ |
| | 10 | $7.9600E - 01$ | $3.3258E - 08$ | $8.4260E - 01$ |
| | 30 | $3.0844E + 00$ | $2.5108E - 06$ | $2.3711E + 00$ |
| PSO | 5 | $6.6300E - 02$ | $0.0000E + 00$ | $2.5240E - 01$ |
| | 10 | $1.7267E + 00$ | $0.0000E + 00$ | $1.1662E + 00$ |
| | 30 | $2.9001E + 01$ | $1.7927E + 01$ | $8.5603E + 00$ |
| ABC | 5 | $0.0000E + 00$ | $0.0000E + 00$ | $0.0000E + 00$ |
| | 10 | $0.0000E + 00$ | $0.0000E + 00$ | $0.0000E + 00$ |
| | 30 | $1.0658E - 15$ | $0.0000E + 00$ | $4.2397E - 15$ |

Table 3: Results of the Rosenbrock function.

| Algorithm | Dimension | Mean | Best | SD |
|---|---|---|---|---|
| SLO | 5 | $1.3325E + 00$ | $1.6163E - 08$ | $8.7800E - 01$ |
| | 10 | $1.6880E + 01$ | $6.6976E + 00$ | $2.3646E + 01$ |
| | 30 | $1.6123E + 04$ | $2.5639E + 03$ | $9.1648E + 03$ |
| GA | 5 | $5.0800E - 02$ | $9.2000E - 03$ | $2.0100E - 02$ |
| | 10 | $7.0620E - 01$ | $1.0590E - 01$ | $5.6060E - 01$ |
| | 30 | $2.3368E + 01$ | $7.5000E - 02$ | $2.2251E + 01$ |
| PSO | 5 | $2.2040E - 01$ | $3.6080E - 04$ | $3.9500E - 01$ |
| | 10 | $2.6053E + 00$ | $3.9800E - 02$ | $1.3755E + 00$ |
| | 30 | $4.1055E + 01$ | $1.1177E + 01$ | $2.7793E + 01$ |
| ABC | 5 | $5.1300E - 02$ | $4.9000E - 03$ | $5.6700E - 02$ |
| | 10 | $5.2200E - 02$ | $1.9000E - 03$ | $5.1100E - 02$ |
| | 30 | $6.0700E - 02$ | $3.6401E - 04$ | $7.5300E - 02$ |

*4.2.3. ABC Settings.* The control parameters of the ABC algorithm are as follows: the maximum number of cycles is equal to the maximum number of generation and the colony size is equal to the population size, that is, 50, as presented in [6]. The percentage of onlooker bees was 50% of the colony, the employed bees were 50% of the colony, and one bee was selected as the scout bee. The increase in the number of scouts encourages the exploration because the increase in onlookers for a food source increases exploitation.

*4.2.4. SLO Settings.* In SLO, each dimension is divided into two equal parts, and thus, $2^D$ patches are generated. In each patch, the initial population of ladybirds is set to 20. The

Table 4: Results of the Ackley function.

| Algorithm | Dimension | Mean | Best | SD |
|---|---|---|---|---|
| SLO | 5 | $-8.8818E-16$ | $-8.8818E-16$ | $0.0000E+00$ |
| | 10 | $9.3400E-02$ | $2.6645E-15$ | $3.6120E-01$ |
| | 30 | $9.2670E-01$ | $6.2172E-15$ | $2.3272E+00$ |
| GA | 5 | $1.9147E-05$ | $8.7429E-07$ | $1.4740E-05$ |
| | 10 | $2.8850E-05$ | $4.1240E-06$ | $1.2182E-05$ |
| | 30 | $7.6502E-05$ | $5.3960E-05$ | $1.2228E-05$ |
| PSO | 5 | $2.0724E-15$ | $-8.8818E-16$ | $1.3467E-15$ |
| | 10 | $3.6119E-15$ | $2.6645E-15$ | $1.5979E-15$ |
| | 30 | $1.7064E-08$ | $1.5857E-09$ | $2.1073E-08$ |
| ABC | 5 | $2.6645E-15$ | $2.6645E-15$ | $0.0000E+00$ |
| | 10 | $6.9278E-15$ | $2.6645E-15$ | $2.1681E-15$ |
| | 30 | $4.4154E-13$ | $1.6964E-13$ | $3.0035E-13$ |

Table 5: Results of the Schwefel function.

| Algorithm | Dimension | Mean | Best | SD |
|---|---|---|---|---|
| SLO | 5 | $3.2505E+02$ | $6.3638E-05$ | $1.7285E+02$ |
| | 10 | $1.2493E+03$ | $7.5042E+02$ | $2.7913E+02$ |
| | 30 | $5.8700E+03$ | $4.7145E+03$ | $7.6059E+02$ |
| GA | 5 | $2.0752E+03$ | $2.0752E+03$ | $5.2257E-11$ |
| | 10 | $4.1504E+03$ | $4.1504E+03$ | $9.4875E-11$ |
| | 30 | $1.2451E+04$ | $1.2451E+04$ | $7.3221E-10$ |
| PSO | 5 | $2.9610E+02$ | $6.3638E-05$ | $1.3104E+02$ |
| | 10 | $5.6719E+02$ | $2.3688E+02$ | $1.6709E+02$ |
| | 30 | $2.8564E+03$ | $1.5989E+03$ | $4.0442E+02$ |
| ABC | 5 | $6.3638E-05$ | $6.3638E-05$ | $1.9653E-14$ |
| | 10 | $1.2728E-04$ | $1.2728E-04$ | $5.2425E-14$ |
| | 30 | $3.1000E-03$ | $3.8183E-04$ | $1.4200E-02$ |

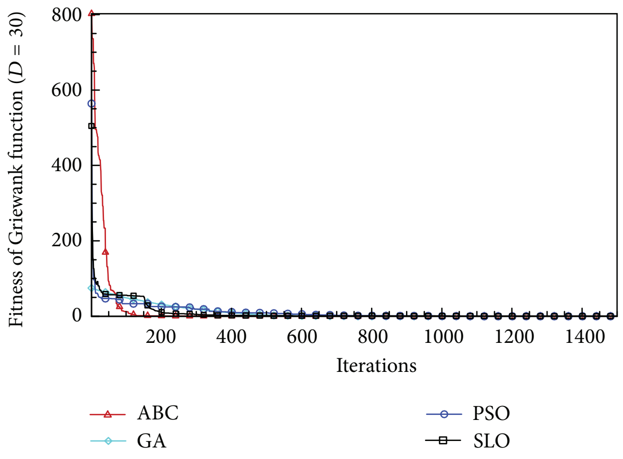


Figure 6: Convergence characteristics of the Griewank function with $D = 30$.

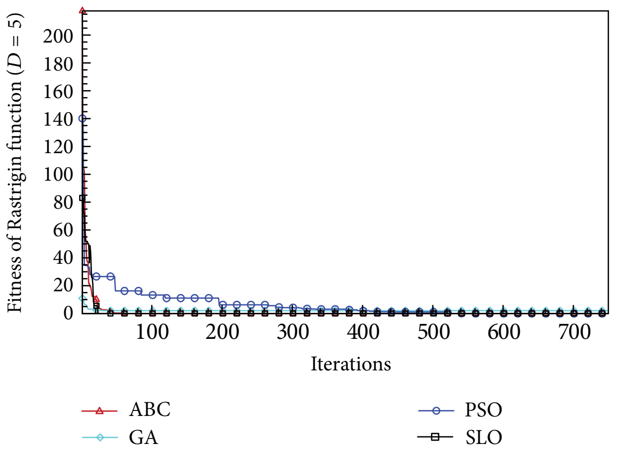


Figure 7: Convergence characteristics of the Rastrigin function with $D = 5$.

parameter *limit* is 100 and $w$ is 1; that is, after 100 cycles of search, if a position in a patch cannot be improved, then it will be abandoned and a new position will be produced in the neighborhood of $g$best. The parameter $c$ in (2) and (4) decreases linearly from 10 to 2. The sideward vibration $\varepsilon_1$ is Rand $*$ $10e-4$ and $\varepsilon_2$ is Rand $*$ $10e-8$.

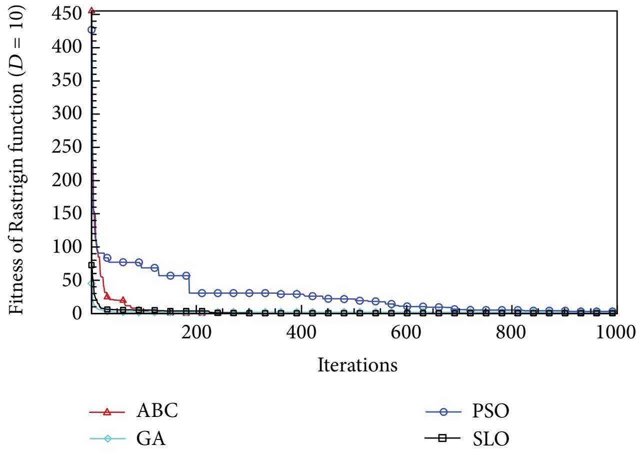


Figure 8: Convergence characteristics of the Rastrigin function with $D = 10$.

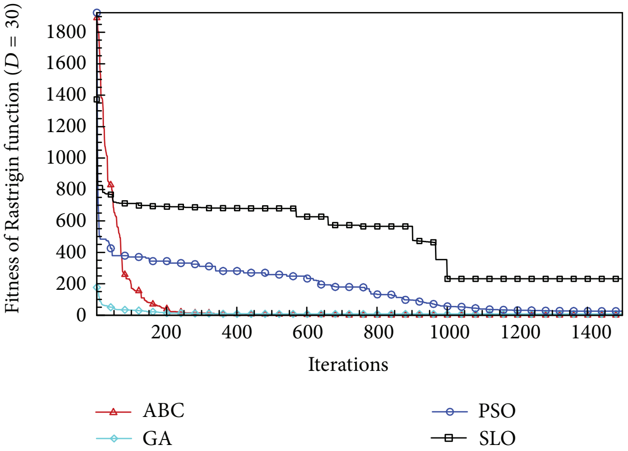


Figure 9: Convergence characteristics of the Rastrigin function with $D = 30$.

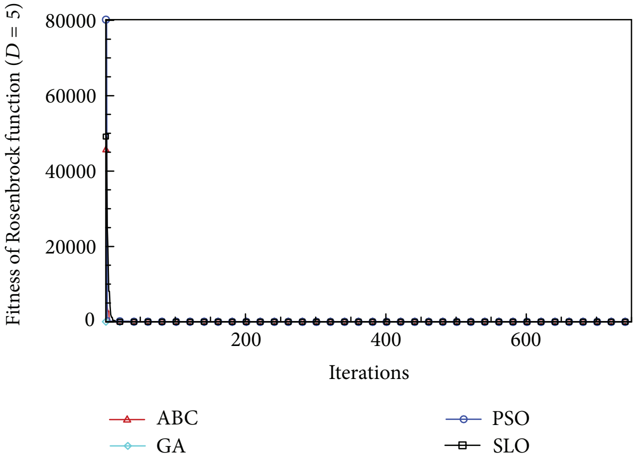


Figure 10: Convergence characteristics of the Rosenbrock function with $D = 5$.

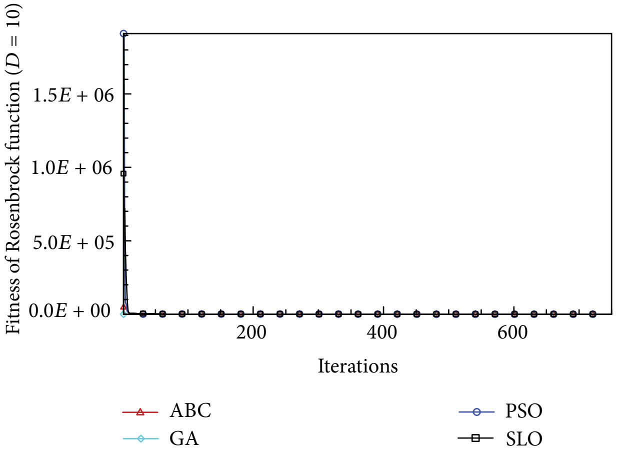


Figure 11: Convergence characteristics of the Rosenbrock function with $D = 10$.

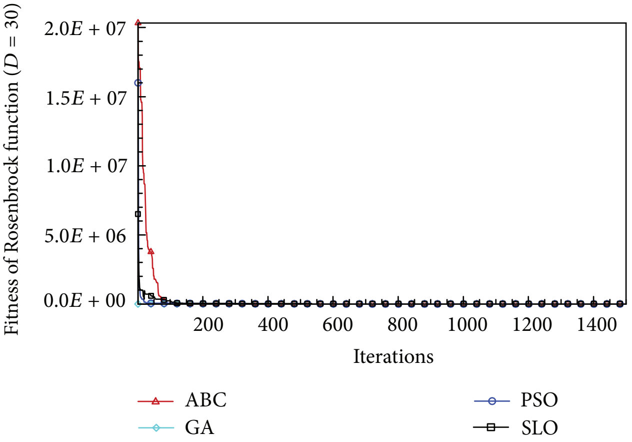


Figure 12: Convergence characteristics of the Rosenbrock function with $D = 30$.

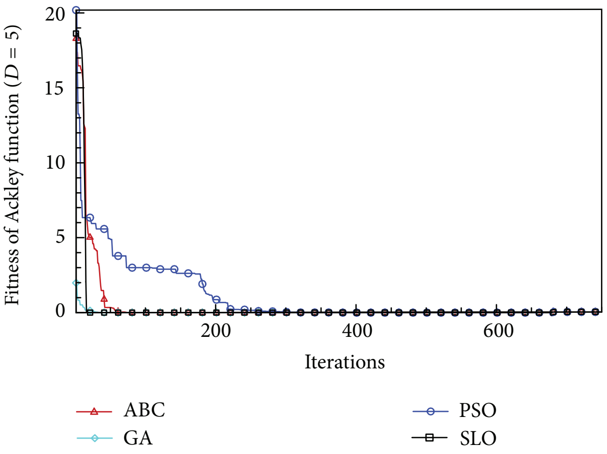


Figure 13: Convergence characteristics of the Ackley function with $D = 5$.

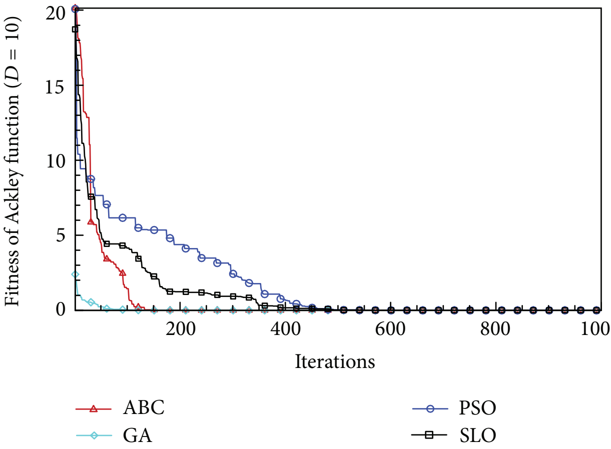


FIGURE 14: Convergence characteristics of the Ackley function with $D = 10$.

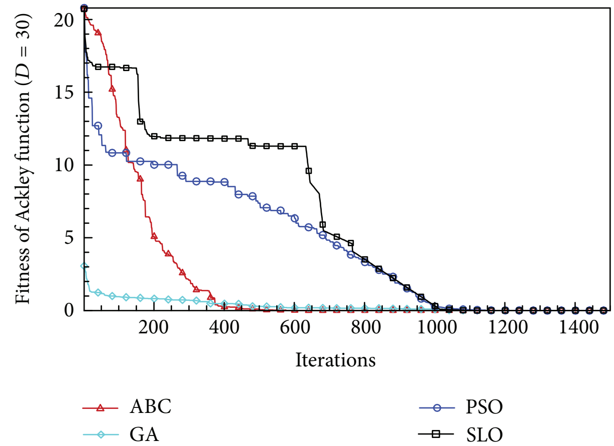


FIGURE 15: Convergence characteristics of the Ackley function with $D = 30$.

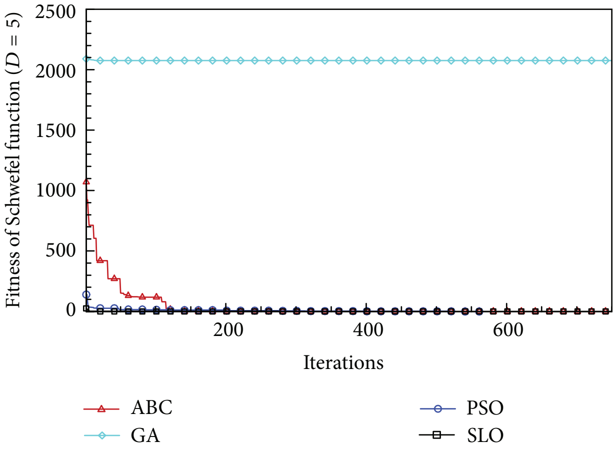


FIGURE 16: Convergence characteristics of the Schwefel function with $D = 5$.

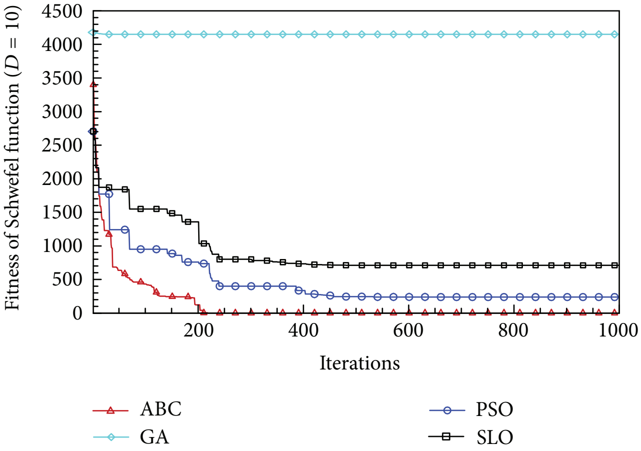


FIGURE 17: Convergence characteristics of the Schwefel function with $D = 10$.

*4.3. Results and Discussion.* In this paper, all the experiments were repeated 30 times with different random seeds. The best and mean function values of the solutions found using the algorithms for different dimensions were recorded. Tables 1, 2, 3, 4, and 5 present the mean, best, and standard deviations of the function values obtained using SLO, GA, PSO, and ABC with $D = 5$, $D = 10$ and $D = 30$. Figures 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, and 18 show the convergence characteristics in terms of the fitness value of each algorithm for each test function.

According to the best function values obtained using the different algorithms with $D = 5$, the SLO can find the global optimization solution with values close to the theoretical solution and has the same search ability as PSO. Many literatures have pointed out that larger population size and large number of generations increase the likelihood of obtaining a global optimum solution. Thus, the performance of PSO with a swarm size of 50 is better than that with a swarm size of 20. According to the experiments in our paper, SLO with a small population of 20 was able to find the global optimization solution with values close to the theoretical solution. This indicates the proposed SLO algorithm has the ability to find the best solution with a comparatively small population size. Based on the mean results of all experiments, the proposed SLO has better performance than GA for Griewank function, Ackley function, and Schwefel function. However, when dimension is 30, the result of SLO is no better than that of PSO and ABC. Comparing the convergence graphs, SLO converged faster and performed better compared with GA.

From the results, we can see that the SLO does not obtain better result along with the growing dimensions. Considering the No Free Lunch Theorem [41], if we compare two searching algorithms with all possible functions, the performance of any two algorithms will be, on average, the same. As a result, when an algorithm is evaluated, we must look for the kind of problems where its performance is good, in order to characterize the type of problems for which the algorithm is

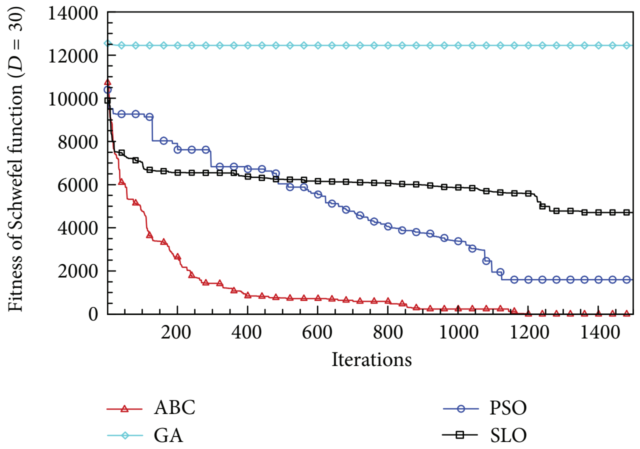


Figure 18: Convergence characteristics of the Schwefel function with $D = 30$.

suitable [42]. In this paper, the proposed SLO is suitable for solving optimization problems with lower dimensions.

## 5. Conclusion

This paper investigated the foraging behaviors of seven-spot ladybirds and proposed a novel biologically inspired metaheuristic algorithm called SLO. The SLO, GA, PSO, and ABC algorithms were tested on five numerical benchmark functions with multimodality to validate the performance of SLO. The simulated results show that SLO has the ability to find the best solution and is suitable for solving optimization problems with lower dimensions. In this paper, the ABC algorithm outperformed all other algorithms, but according to the No Free Lunch Theorem [41], "any elevated performance over one class of problems is offset by performance over another class." Future studies will focus on improving the SLO.

## Acknowledgments

The authors are grateful to the editor and the anonymous referees for their insightful and constructive comments and suggestions, which have been very helpful for improving this paper. This research was supported by the National Natural Science Foundation of China (Grant no. 51375389) and the National High Technology Research and Development Program of China (863 Program) no. 2011AA09A104.

## References

[1] D. T. Pham and D. Karaboga, *Intelligent Optimisation Techniques*, Springer, New York, NY, USA, 2000.

[2] F. Glover and G. A. Kochenberger, *Handbook of Metaheuristics*, Kluwer Academic, Boston, Mass, USA, 2003.

[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Lansing, Mich, USA, 1975.

[4] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.

[5] R. C. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.

[6] 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.

[7] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, Coimbatore, India, December 2009.

[8] A. Mucherino and O. Seref, "Monkey search: a novel metaheuristic search for global optimization," in *Proceedings of the Conference on Data Mining, Systems Analysis, and Optimization in Biomedicine*, pp. 162–173, Gainesville, Fla, USA, March 2007.

[9] X. S. Yang, "Firefly algorithms for multimodal optimization," *Stochastic Algorithms: Foundations and Applications*, vol. 5792, pp. 169–178, 2009.

[10] A. Ahrari and A. A. Atai, "Grenade explosion method—a novel tool for optimization of multimodal functions," *Applied Soft Computing Journal*, vol. 10, no. 4, pp. 1132–1140, 2010.

[11] S. C. Chu, P. W. Tsai, and J. S. Pan, "Cat swarm optimization," in *PRICAI 2006: Trends in Artificial Intelligence*, vol. 4099 of *Lecture Notes in Computer Science*, pp. 854–858, Springer, Guilin, China, 2006.

[12] B. Alatas, "Acroa: artificial chemical reaction optimization algorithm for global optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13170–13180, 2011.

[13] M. Kapanoglu and W. A. Miller, "An evolutionary algorithm-based decision support system for managing flexible manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 6, pp. 529–539, 2004.

[14] N. Mansour, H. Tabbara, and T. Dana, "A genetic algorithm approach for regrouping service sites," *Computers and Operations Research*, vol. 31, no. 8, pp. 1317–1333, 2004.

[15] Z. Lian, X. Gu, and B. Jiao, "A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan," *Applied Mathematics and Computation*, vol. 175, no. 1, pp. 773–785, 2006.

[16] L. Barcos, V. Rodríguez, M. J. Álvarez, and F. Robusté, "Routing design for less-than-truckload motor carriers using ant colony optimization," *Transportation Research E*, vol. 46, no. 3, pp. 367–383, 2010.

[17] G. N. Ramos, Y. Hatakeyama, F. Dong, and K. Hirota, "Hyperbox clustering with ant colony optimization (HACO) method and its application to medical risk profile recognition," *Applied Soft Computing Journal*, vol. 9, no. 2, pp. 632–640, 2009.

[18] J. P. Hamiez and J. K. Hao, "Using solution properties within an enumerative search to solve a sports league scheduling problem," *Discrete Applied Mathematics*, vol. 156, no. 10, pp. 1683–1693, 2008.

[19] M. Tamer Ayvaz, "Application of harmony search algorithm to the solution of groundwater management models," *Advances in Water Resources*, vol. 32, no. 6, pp. 916–924, 2009.

[20] P. Charbonneau, "Genetic algorithms in astronomy and astrophysics," *Astrophysical Journal*, vol. 101, no. 2, pp. 309–334, 1995.

[21] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[22] D. Srinivasan and T. H. Seow, "Particle swarm inspired evolutionary algorithm (ps-ea) for multiobjective optimization problems," in *Proceedings of the Congress on Evolutionary Computation*, pp. 2292–2297, Canberra, Australia, 2003.

[23] R. Hassan, B. Cohanim, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pp. 1–13, Austin, Tex, USA, April 2005.

[24] B. Akay and D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012.

[25] I. Hodek, G. Iperti, and M. Hodkova, "Long-distance flights in Coccinellidae (Coleoptera)," *The European Journal of Entomology*, vol. 90, no. 4, pp. 403–414, 1993.

[26] A. F. G. Dixon, *Insect Predator-Prey Dynamics: Ladybird Beetles and Biological Control*, Cambridge University Press, New York, NY, USA, 2000.

[27] J. L. Hemptinne, M. Gaudin, A. F. G. Dixon, and G. Lognay, "Social feeding in ladybird beetles: adaptive significance and mechanism," *Chemoecology*, vol. 10, no. 3, pp. 149–152, 2000.

[28] J. L. Hemptinne, G. Lognay, C. Gauthier, and A. F. G. Dixon, "Role of surface chemical signals in egg cannibalism and intraguild predation in ladybirds (Coleoptera: Coccinellidae)," *Chemoecology*, vol. 10, no. 3, pp. 123–128, 2000.

[29] J. Pettersson, V. Ninkovic, R. Glinwood, M. A. Birkett, and J. A. Pickett, "Foraging in a complex environment-semiochemicals support searching behaviour of the seven spot ladybird," *The European Journal of Entomology*, vol. 102, no. 3, pp. 365–370, 2005.

[30] V. Ninkovic, S. Al Abassi, and J. Pettersson, "The influence of aphid-induced plant volatiles on ladybird beetle searching behavior," *Biological Control*, vol. 21, no. 2, pp. 191–195, 2001.

[31] N. Suzuki and T. Ide, "The foraging behaviors of larvae of the ladybird beetle, Coccinella septempunctata L., (Coleoptera: Coccinellidae) towards ant-tended and non-ant-tended aphids," *Ecological Research*, vol. 23, no. 2, pp. 371–378, 2008.

[32] A. Vantaux, O. Roux, A. Magro, and J. Orivel, "Evolutionary perspectives on myrmecophily in ladybirds," *Psyche*, vol. 2012, Article ID 591570, 7 pages, 2012.

[33] M. P. Hassell and T. R. E. Southwood, "Foraging strategies of insects," *Annual Review of Ecology and Systematics*, vol. 9, pp. 75–98, 1978.

[34] I. Hodek, S. Chakrabarti, and M. Rejmanek, "The effect of prey density on food intake by adult Cheilomenes sulphurea [Col.: Coccinellidae]," *BioControl*, vol. 29, no. 2, pp. 179–184, 1984.

[35] A. Ferran and A. F. G. Dixon, "Foraging behaviour of ladybird larvae (Coleoptera: Coccinellidae)," *The European Journal of Entomology*, vol. 90, no. 4, pp. 383–402, 1993.

[36] J. L. Hemptinne, A. F. G. Dixon, and J. Coffin, "Attack strategy of ladybird beetles (Coccinellidae): factors shaping their numerical response," *Oecologia*, vol. 90, no. 2, pp. 238–245, 1992.

[37] P. Kindlmann and A. F. G. Dixon, "Optimal foraging in ladybird beetles (Coleoptera: Coccinellidae) and its consequences for their use in biological control," *European Journal of Entomology*, vol. 90, no. 4, pp. 443–450, 1993.

[38] N. Minoretti and W. W. Weisser, "The impact of individual ladybirds (Coccinella septempunctata, Coleoptera: Coccinellidae) on aphid colonies," *The European Journal of Entomology*, vol. 97, no. 4, pp. 475–479, 2000.

[39] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 81–86, Seoul, Republic of Korea, May 2001.

[40] D. Ortiz-Boyer, C. Hervás-Martínez, and N. García-Pedrajas, "CIXL2: a crossover operator for evolutionary algorithms based on population features," *Journal of Artificial Intelligence Research*, vol. 24, pp. 1–48, 2005.

[41] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[42] F. Valdez and P. Melin, "Comparative study of particle swarm optimization and genetic algorithms for complex mathematical functions," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 2, no. 1, pp. 43–51, 2008.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 369209, 13 pages
http://dx.doi.org/10.1155/2013/369209

*Research Article*

# A Novel Artificial Bee Colony Approach of Live Virtual Machine Migration Policy Using Bayes Theorem

**Gaochao Xu,[1,2] Yan Ding,[1,2] Jia Zhao,[1,2] Liang Hu,[1,2] and Xiaodong Fu[1,2]**

[1] *College of Computer Science and Technology, Jilin University, Changchun, Jilin 130000, China*
[2] *Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130000, China*

Correspondence should be addressed to Jia Zhao; zhaiyj049@sina.com

Received 30 September 2013; Accepted 17 November 2013

Academic Editors: T. Chen, Q. Cheng, and J. Yang

Green cloud data center has become a research hotspot of virtualized cloud computing architecture. Since live virtual machine (VM) migration technology is widely used and studied in cloud computing, we have focused on the VM placement selection of live migration for power saving. We present a novel heuristic approach which is called PS-ABC. Its algorithm includes two parts. One is that it combines the artificial bee colony (ABC) idea with the uniform random initialization idea, the binary search idea, and Boltzmann selection policy to achieve an improved ABC-based approach with better global exploration's ability and local exploitation's ability. The other one is that it uses the Bayes theorem to further optimize the improved ABC-based process to faster get the final optimal solution. As a result, the whole approach achieves a longer-term efficient optimization for power saving. The experimental results demonstrate that PS-ABC evidently reduces the total incremental power consumption and better protects the performance of VM running and migrating compared with the existing research. It makes the result of live VM migration more high-effective and meaningful.

## 1. Introduction

VM technology [1, 2], one of the most important technologies in cloud computing, is not only a way to implementing cloud computing such as Infrastructure as a Service (IaaS) [3] architecture but also for the embody of the cloud computing idea, whereas live VM migration technology, which is widely used for the maintenance management in virtualized cloud computing data centers, is the representative of the VM technologies. When a VM needs migrating from source host to target host for some goal or several goals, generally the migration target of a VM is chosen randomly as long as the host can accommodate it and then one can automatically or manually move the VM to a target host. It is obvious that the way to randomly choose a target host for a live VM migration which some event has aroused and has more than one available target host meeting the requirements of that event is not efficient in all respects.

Nowadays, power consumption of data centers has huge impact on environments. Researchers have been seeking to find effective solutions to minimize power consumption of data centers while keeping the desired quality of service. On the background of low-carbon world and cloud computing era, researchers have proposed the field of green cloud computing based on cloud computing and virtualization. It aims at reducing power consumption in cloud computing data centers. In this paper, we focus on live VM migration policy based on green data center. In a cloud data center, there are always some VMs needing to be migrated for some reasons. Generally speaking, the migrant VM has many available target hosts. However, only one target host is most suitable for the VM in order to minimize the total incremental power consumption in cloud data center. To achieve green cloud data center, in the search direction of VM provisioning the problem is similar to that of live VM migration. Both are to find out the optimal target host.

Many papers have presented some heuristic approaches to find optimal solutions aiming to minimize power consumption. The basic idea is that according to the current situation and history of a cloud data center, the controllers have searched for a best policy by using their proposed approaches. The problems of convergence and local optimization have been challenging the research direction. On the other hand, we know that a data center does not have abilities in predicting the size and type of the next workloads. Therefore, the optimal policy which the proposed approaches have found out in a short-term is not necessarily the optimal solution in a long-term. In a word, the global best which of some VM the proposed approaches have found out in an algorithm cycle may be a local best in a long-term process. Thus, the probability methods and ideas should be introduced into these heuristic approaches aiming at this kind of problems. Besides, as the capability that the current random migration policy and optimal migration policy adapt to a dynamic cloud environment is not excellent enough, they may cause many failure events of live VM migration in a real and dynamic cloud environment. To address these problems, this paper presents a novel heuristic approach PS-ABC, which designs and employs the improved artificial bee colony approache and utilizes the binary search idea and the Boltzmann selection policy to achieve a more efficient power saving optimization. Also, the Bayes theorem is introduced into the proposed PS-ABC approach and makes it have capability in fast converging to global optimal solution while improving the performance of live VM migration. That is, the failure rate of VM migration is decreased. Compared to the random approach and other existing optimization approaches, the proposed PS-ABC has reduced more power consumption and failure events of live VM migration to contribute to achieving better green cloud data centers.

The rest of the paper is organized as follows. In Section 2, we present the related work related to our proposed approach aiming to green cloud data center and the reasonable prerequisites are shown clearly. In Section 3, the analysis of the problem proposed in this paper and its formulation are given. In Section 4, the algorithm and implementation of PS-ABC are introduced in detail. In Section 5, the experimental results and analysis on CloudSim platform are given. Finally, in Section 6, we summarize the full paper and future work is put forward.

## 2. Related Work

As far as we know, the proposed problem which refers to finding a fit target host for a live VM migration according to the standard of minimizing the increment power consumption has not been widely researched in the related fields. However, most researchers have focused on some problems which are similar to the proposed problem in this paper. Also, some researchers have focused on the direction that aims to other problems of cloud computing and also to minimize the incremental power consumption but utilizes the technology of live VM migration. Similarly, there are also some researchers studying the direction which utilizes live VM migration to move these VMs in order to fulfill the requirement of performance and workload limitation while minimizing the power drawn by a cloud data center. In fact, most problems of them are just to find an optimal host for each VM, which will be migrated or be created, under the target of minimizing the power drawn by a cloud data center. Thus, the related work of the kind of problems of green cloud data centers will be discussed briefly in this section.

Rusu et al. in [4] have presented a cluster-wide QoS-aware technique that dynamically reconfigures the cluster to reduce power consumption during periods of reduced load. The proposed system consists of two important components, namely, front end manager and a local manager. While the front end manager finds the servers which should be turned on or off in terms of a given system load, the local manager will utilize Dynamic Voltage and Frequency Scaling (DVFS) technique to conserve power. The main shortage of the approach is the on/off policy. It relies on the table of values and needs computing offline. However, the system does not make use of server consolidation through VM migration and thus its on/off policy may not be much effective.

Srikantaiah et al. [5] have investigated the problem of dynamic consolidation of applications serving small stateless requests in data centers to minimize the power consumption. They modeled the problem as a multidimensional bin packing problem. However, the proposed model does not describe the degradation of performance due to the consolidation. Besides, the power drawn may rely on a particular set of applications combined on a computer node. A heuristic for the defined bin packing problem is proposed by the authors. The heuristic is based upon the idea of minimizing the sum of the current allocations' Euclidean distances to the optimal point at each server. The application workload will be allocated to a server using the proposed heuristic since a request to execute a new application is received. Without the sufficient capacity of all active, the system will switch on a new server while reallocating all the applications using the same heuristic in an arbitrary order. The proposed approached is fit for heterogeneous environments; however, it has several shortcomings. First, the approach assumes that all applications' resource requirements are known in advance and constant. Second, performance and power overhead, which the authors do not take into account, is caused by migration of state-full applications between nodes. The frequent switching servers on/off also generate significant costs which are not negligible for a real-world system.

Verma et al. [6] have contributed power and migration cost-aware application placement by exploiting the power management capabilities of virtualization. The authors have designed a new application (virtual machines) placement architecture called pMapper. It consists of three major parts, namely, a performance manager to dynamically resize the VM, a power manager for CPU throttling, and a migration manager to identify the target host for migration using a knowledge base. They have expounded that for power-aware scheduling approaches, estimates of power values are not required, and only if the scheduling algorithm has abilities in finding out which server minimizes the incremental increase in total power owing to the new VM being placed, it can place the given VM to appropriate host. In pMapper, two algorithms are implemented. One is First Fit Decreasing (FFD) by which more power-efficient servers are utilized first without balancing the load. The other is incremental First Fit Decreasing (iFFD) which considers the fixed target utilization of each server and achieves server consolidation by live VM migration. The proposed pMapper architecture minimizes power and migration costs with ensuring the performance. Our approach is based on a heuristic approach which exploits the concept of minimizing total increase in the incremental power due to the new VM migrations. The proposed architecture is simple and does not need any knowledge base to achieve significant reduction in the power consumption.

Li et al. in [7] have proposed an approach named EnaCloud, which enables application live placement dynamically with consideration of power efficiency in a cloud platform. In EnaCloud, they use a virtual machine to encapsulate the application, which supports applications scheduling and live migration to minimize the number of running machines, so as to save power. Specially, the application placement is abstracted as a bin packing problem, and a power-aware heuristic algorithm is proposed to get an appropriate solution. In addition, an over-provision approach is presented to deal with the varying resource demands of applications. However, the over-provision approach has risk to optimize this problem. It may cause more cost in order to reduce less cost with a certain probability.

Jeyarani et al. [8] have proposed self-adaptive particle swarm optimization (SAPSO) for efficient virtual machine provisioning in cloud aimed at the following: when mapping a set of VM instances onto a set of servers from a dynamic resource pool, the total incremental power drawn upon the mapping is minimal and does not compromise the performance objectives. The advantage of the proposed solution is obvious. It has focused on not only improving the performance of workload facilitating the cloud consumers but also developing the power efficient data center management to facilitate cloud providers. However, the approach still may be inefficient and cause some additional events and costs from a long-term perspective as it does not take the future workload into account. Our proposed algorithm PS-ABC is a heuristic approach which is based on ABC, one of swarm intelligence algorithms and introduces the Boltzmann selection idea into it.

In this paper, PS-ABC has a prerequisite. We know that PS-ABC is to find the target host of each VM from all $m$ hosts for the $n$ migrant VMs. We assume that each VM's target host found by PS-ABC will not be the host which the VM is moved out from. The approach provides service to live VM migration aiming to green cloud data center. Thus, the fact that the VM should be moved out for some reason is the premise of our approach. Talking objectively, the prerequisite is justified from a certain perspective. Since a VM needs to be migrated from its source host, its candidate hosts will not include its source host. Otherwise, it does not need a migration event. Further, this matter that a host both needs move out VMs and has ability in receiving VMs within a time window $\Delta t$ is impossible and nonobjective. It can be seen that for all the migrant VMs, the hosts each of which is the source host of some migrant VM will not be the target hosts. Therefore, the proposed prerequisite is reasonable and does not impact the performance and efficiency of our approach.

## 3. The Proposed Problem and Its Formulation

*3.1. The Proposed Problem.* In IaaS cloud platform, some of the running VMs may be required to be migrated for some reasons. Generally, the target of live VM migration is more than one host. Moreover, the different selections of target hosts will cause the different power consumption. Therefore, a high-efficient power saving placement selection policy to migrate the migrant VMs onto the right fit hosts is necessary.

*3.2. Problem Formulation.* We now formulate the problem of migrating $n$ VMs onto $m$ hosts. Its solution can be represented by an $n$ dimension of solution vector, each element of which denotes the target host of the migrant VM which its location represents. We assume that there are $m$ available hosts in the resource pool and the hosts are heterogeneous and dynamic while using space shared allocation policy. The hosts change their state dynamically according to the load. The problem can be stated as follows. Find a VM-host set Vh of placement selections, such that the total incremental power consumption caused by the migrated VMs onto hosts is minimized, while maximizing the performance by fulfilling the resource requirements of maximum number of VMs. We define a four tuple $S = \{\mathrm{PH}, \mathrm{VM}, \mathrm{PC}, \mathrm{Vh}\}$ for our problem scenario. PH is a set of $m$ available physical hosts denoted by $\mathrm{PH}(m, t) = \{\mathrm{PH}_1, \mathrm{PH}_2, \mathrm{PH}_3 \ldots, \mathrm{PH}_m\}$, available at migrating start time $t$. VM is a set of $n$ VMs denoted by VM $(n, t, \Delta t) = \{\mathrm{VM}_1, \mathrm{VM}_2, \mathrm{VM}_3, \ldots, \mathrm{VM}_n\}$ accumulated within a time window $\Delta t$. $\mathrm{PC}(m, t) = \{\mathrm{PC}_1, \mathrm{PC}_2, \ldots, \mathrm{PC}_m\}$ is the power consumption by the $m$ physical hosts in resource pool. The problem is multimodal, having more than one placement selection which meets the performance constraints of the VM requests. Therefore, our goal is to find all of $o$ candidate placement selections which maximize the performance and then to find which one minimizes the power consumption among them. To fulfill performance constraints, a metric denoted by $\eta_r$ representing resource fulfillment requirement is defined as follows:

$$\eta_r = \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_i^j, \quad i \in \{1, 2, 3, \ldots, n\}, \qquad (1)$$

$$j \in \{1, 2, 3, \ldots, m\}, \quad r \in \{1, 2, 3, \ldots, s\},$$

where $\phi_i^j$ denotes the placement selection of $i$th VM on the $j$th host and is defined as follows. $s$ denotes the total number of placement selections.

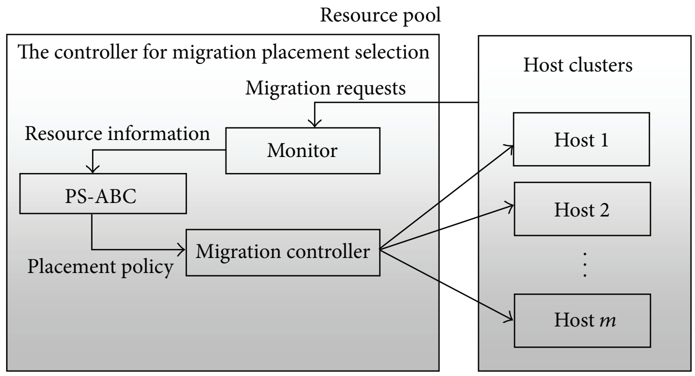


Figure 1: The view of PS-ABC's architecture.

Consider

$$\phi_i^j = \begin{cases} 1 \\ \quad \text{if VM}_i \text{ has been allocated to} \\ \quad \text{Host}_j \text{ and } rcr\text{VM}_i \leq acr\text{Host}_j, \\ \quad i \in \{1, 2, 3, \ldots, n\} \\ 0 \\ \quad \text{if VM}_i \text{ has been allocated to} \\ \quad \text{Host}_j \text{ and } rcr\text{VM}_i > acr\text{Host}_j, \\ \quad j \in \{1, 2, 3, \ldots, m\} \\ \text{invalid} \\ \quad \text{if VM}_i \text{ has not been allocated to} \\ \quad \text{Host}_j. \end{cases} \tag{2}$$

In (2), $rcr\text{VM}_i$ denotes the minimum computing resource requirements of $i$th VM and $acr\text{Host}_j$ denotes the available computing resource of $j$th host. Out of $s$ placement selections, $o$ placement selections with maximum $\eta_r$ values give the best performance and they are represented as $\text{Vh}[m, n, o, t]$.

In fact, $\text{Vh}[m, n, o, t]$ represents $o$ solution vectors with maximum $\eta_r$ values. The second metric is based on the power consumption. The migration of successive VMs is represented as $\text{Vh}[m, n, r, t + t_0(k)]$ where $r$ represents any one of $o$ placement selections and $k$ is an integer increasing with successive migrating, representing a stage. We can understand like that if $k$ is 3, the approach will migrate the third VM to the host which is denoted at the third location of the $r$ vector. The $o$ placement selections at stage $k$ is represented as $\text{Vh}[m, n, r, t + t_0(k)] r \in \{1, 2, 3, \ldots, o\}$ and their corresponding power consumption is represented as $\xi_k^r$. Its meaning is that, according to $r$ placement vector after the system has migrated the $k$th VM to its target host, the total power consumption by the cloud data center is $\xi_k^r$. The meaning of $\xi_{k-1}^r$ can be imaged. In this paper, these parameters of $\xi_k^r$ can be obtained by using simulation platform in the experiment. Now the incremental power consumption due to migrating $\text{Vh}[m, n, r, t + t_0(k)]$ with respect to previous migration stage $\text{Vh}[m, n, r, t + t_0(k-1)]$ is defined by

$$\Delta P = \left(\xi_k^r\right) - \left(\xi_{k-1}^r\right), \quad r \in \{1, 2, 3, \ldots, o\}. \tag{3}$$

For better power saving, the following $\delta P$ is minimized to get the optimal solution and it is denoted as follows:

$$\delta P = \sum_{k=1}^{n} \left(\left(\xi_k^r\right) - \left(\xi_{k-1}^r\right)\right), \quad r \in \{1, 2, 3, \ldots, o\}. \tag{4}$$

Therefore, the proposed approach maximizes $\eta_r$ for better performance requirements and then minimizes $\delta P$ for power efficiency.

## 4. Heuristic Methodology for Efficient Placement Selections of Live VM Migration

*4.1. The Proposed System Architecture.* Figure 1 depicts the proposed architecture for cloud environment. It shows the position of the controller PS-ABC for migration placement selections and its interaction with other entities [9, 10]. In a time window $\Delta t$, the Monitor gets the requests of live VM migration and is updated with the available number of computing resource such as CPUs, memory, and storage as well as power consumption. At the end of $\Delta t$, the Monitor transfers the information to the controller. The controller generates the placement policy by using the proposed approach and obtained information. Then, it transfers the policy to the migration controller which controls and executes live migration of the VMs. The VMs are moved onto their target hosts eventually.

*4.2. Solution Representation.* In order to design an efficient ABC-based approach for finding the optimal solution vector of the target hosts of all migrant VMs in a time window $\Delta t$, the primary problem is the solution representation as it represents a direct relationship between the problem domain and the food sources in ABC. During the applying of a ABC-based approach, we know that a food source is denoted as a solution of the specific problem. Here, there are $n$ VMs to be migrated into $m$ physical hosts. So, the proposed problem is an $n$ dimensional problem, leading to the fact that the food sources are represented as $n$ dimensional solution vectors. Each dimension has a discrete set of possible placement selection limited to $m$. A solution vector called a food source is denoted by $x_i^k = (x_{i1}^k, x_{i2}^k, \ldots, x_{ij}^k, \ldots, x_{iD}^k)$, where the $x_{ij}^k$

value represents the Number of target host of the $j$th migrant VM of the $i$th possible solution vector in $k$th generation.

*4.3. The Algorithm of PS-ABC.* The proposed PS-ABC approach is a novel heuristic live VM migration policy, which is used for the target location selection of live VM migration. It is based on the improved ABC approach, in which firstly we present the uniform random initialization idea to improve the initialization process of food sources and thus to make the PS-ABC approach have a better global search ability at the very beginning; secondly, we utilize the binary search idea to improve the expression, by using which each employed bee produces a candidate food position, for increasing the neighborhood searching efficiency of the employed bees and thus to achieve the high-efficient global search and local convergence; thirdly the PS-ABC approach employs the Boltzmann selection policy to achieve the probability selection of onlookers and thus to avoid prematurity and make the PS-ABC approach have the self-adaptive selective pressure adjustment so as to achieve having the better global exploration ability in the early stage and the better exploitation ability in the later stage; finally we introduce the Bayes Theorem into the proposed PS-ABC approach to optimize its convergence efficiency and improve its accuracy.

*4.4. The Implementation of PS-ABC.* In this section, we describe the specific process of PS-ABC. Details are as follows.

*4.4.1. Monitor the Requests of Live VM Migration.* The proposed controller gets periodically the requests of live VM migration and the information of computing resource. The time is denoted as $\Delta t$. In other words, the proposed PS-ABC is executed once at intervals of $\Delta t$. In a time window $\Delta t$, the number of VM migration requests obtained is the dimension number of a solution vector of the problem to be resolved by the proposed approach. After a $\Delta t$ ends, the monitor transmits the relevant information required by the proposed approach to the module used for running the algorithm.

*4.4.2. Initialize the Parameters and Food Sources.* The PS-ABC module receives all required information and begins running the algorithm. First, the parameters required by the approach are initialized such as the number FSN of food sources. For each food source, there is only one employed bee. That is, the number of employed bees is equal to FSN. In ABC, the number of onlookers is equal to the number FSN of employed bees. The maximum number of iterations is MNI. The dimension of the problem to be solved is $D$. The controlling parameter for the ABC process is LIMIT. The parameter LIMIT is used to control when an employed bee abandons its current food source and becomes a scout to have a new random candidate food source. In the proposed PS-ABC approach, the parameter LIMIT does not only play the role but also is taken use of to control when the Bayes theorem works for achieving fast convergence of the algorithm. After initializing the parameters, the PS-ABC begins to generate the initial population of FSN food sources. In the PS-ABC, the initial food sources are not produced at random within the search space but are generated in terms of the proposed uniform random idea. The whole search space is evenly divided into FSN small subspaces. Within each subspace, a food source is generated randomly. As a result, the FSN initial food sources are generated in the whole search space. In this way, the PS-ABC approach has a better global search potential while retaining the traditional randomness. Further, in this case where the added overhead is almost negligible, the performance improvement is noticeable due to the employment of uniform random initialization idea, not only from the perspective of diversity and global search optimization potential but also from the perspective of improving the search efficiency and achieving fast convergence.

*4.4.3. Main Iterative Process of the PS-ABC Algorithm.* After the swarm of FSN employed bees, the swarm of FSN onlookers, the initial population of FSN food sources, and all the relevant parameters are initialized, the PS-ABC begins to perform the first round of iterations. Obviously, each food source has an employed bee. The FSN employed bees concurrently calculate the first fitness value $\eta_r$ of the FSN initial food sources on performance requirements according to the expression, (1) and (2). Then they calculate the second fitness value $\delta P$ of the FSN initial food sources on power consumption and each employed bee keeps its two fitness values in its own memory. Afterwards, these employed bees perform neighbourhood search according the following expression:

$$v_{ij} = \frac{x_{ij} + x_{kj}}{2}, \tag{5}$$

where $k \in \{1, 2, \ldots, \text{FSN}\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. $x_{ij}$ is the current food source which the $i$th employed bee is located at. Although $k$ is determined randomly, it has to be different from $i$. In the classic ABC approach, the expression for the employed bees' neighbourhood search is as follows:

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right), \tag{6}$$

where $\phi_{ij}$ is a random number between −1 and 1. It controls the production of a neighbour food source position around $x_{ij}$ and the modification represents the comparison of the neighbour food positions visually by the bee [11, 12]. In the proposed PS-ABC approach, the parameter $\phi_{ij}$ is not a random number and is set as −1/2 to utilize the binary search and thus to achieve more efficient neighbour search. It also makes the PS-ABC approach have the better exploration and exploitation abilities at the same time to some extent. In our proposed problem, the performance fulfillment is the constraint condition. As a result, the first fitness value $\eta_r$ should be maximized even if the current situation cannot make $\eta_r$ equal to $D$. Under this background, the PS-ABC approach gives each employed bee $AL$ chances to attempt to find the new food source which can make $\eta_r$ equal to $D$ during neighbourhood search. $AL$ is a predetermined controlling parameter in the proposed PS-ABC approach. After that, if an employed bee still cannot find that kind of food source, the employed bee will return to the food source which has the maximum $\eta_r$ value during $AL$ times of neighbourhood search. That is, the result of neighborhood search of that employed bee is the food source which has the maximum $\eta_r$ value during $AL$ times of neighbourhood search. Each employed bee compares the new candidate food source by neighborhood search process with the current food source in the memory. If the $\eta_r$ value of the new food source is larger than that of the old source, it is replaced with the old one in the memory. If the new food source has smaller $\eta_r$ value than the old source, the old one is retained. If the new food source has equal $\eta_r$ value as the old source, the second fitness value $\delta P$ will be compared. If the new food source has equal or smaller $\delta P$ value than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained. Now we define a partial order: $\prec_{\eta_r,\delta P}$ as follows:

$$\begin{aligned} x_\alpha \prec_{\eta_r,\delta P} x_\beta \quad &\text{if } \left(x_\alpha \cdot \eta_r > x_\beta \cdot \eta_r\right) \\ &\text{or } \left(\left(x_\alpha \cdot \eta_r = x_\beta \cdot \eta_r\right), \left(x_\alpha \cdot \delta P < x_\beta \cdot \delta P\right)\right). \end{aligned} \tag{7}$$

It represents that the food source $x_\alpha$ is better than the food source $x_\beta$. After neighborhood search process of the FSN employed bees in the first round of iterations, the PS-ABC approach will compare the first fitness $\eta_r$ values of all employed bees' food sources. The food sources whose $\eta_r$ values are not the maximum of all set their $\delta P$ values to 0.

All these employed bees share their information with onlookers by dancing within the hive and then the onlookers select one of the food sources. The preference of a food source by an onlooker bee depends on the second fitness value $\delta P$ of that food source. As the second fitness value $\delta P$ of the food source decreases, the probability with the preferred source by an onlooker bee increases. In order to dynamically adjust selection pressure during searching the optimal solution, the PS-ABC approach does not utilize the simple roulette wheel selection but introduces the Boltzmann selection policy into the searching process to achieve the selection probability with which onlooker bees choose their food sources to follow. According to Boltzmann selection policy, the probability with the food source located at $x_i$ which will be chosen by an onlooker bee can be expressed as

$$\begin{gathered} p_i = \frac{\exp\left(\mu_i/T\right)}{\sum_{n=1}^{\mathrm{FSN}} \exp\left(\mu_n/T\right)}, \\ T = T_0, \qquad c = 1, \\ T = \left(\alpha^{c-1}\right) T, \quad 1 < c \le \mathrm{MNI}, \end{gathered} \tag{8}$$

where $T$ is the current temperature, $T_0$ is the initial temperature, $\alpha$ is a number between 0 and 1, $c$ is the current number of iterations of PS-ABC, and $\mu_i$ is the reciprocal of the $i$th food source's second fitness value. That is,

$$\mu_i = \frac{1}{\delta P_i}. \tag{9}$$

In order to calculate the probability value $p_i$, we do not directly utilize the fitness value $\delta P_i$ of the $i$th food source but take use of its reciprocal. This is because our proposed problem needs to make the food source with a less power consumption have a larger probability value to be used for onlooker bees' selection. The reasonable transformation is simple and necessary.

After watching the dances of employed bees, an onlooker bee goes to the region of the food source located at $x_i$ by the probability $p_i$ and determines a neighbourhood food source to take its nectar depending on the fitness values. In other words, the onlooker bee selects one of the food sources after making a comparison among the food sources around $x_i$. The position of the selected neighbourhood food source is calculated as expression (6). After the candidate food source position $v_{ij}$ is produced and then evaluated by the onlooker bee, its performance is compared with that of $x_{ij}$. If $v_i \prec_{\eta_r,\delta P} x_i$, then the position $x_i$ of the food source is changed to be $v_i$; otherwise $x_i$ is kept as it is.

After each of all onlooker bees of each employed bee finishes neighbourhood search, the optimal food source so far is recorded in the whole search space. The first round of iterations ends. The iteration number increases by 1. The PS-ABC approach determines whether there exists any food source which has not improved for LIMIT rounds of iterations continuously and the food source is not the best one during LIMIT rounds of iterations. If yes, the food source is abandoned. And its employed bee becomes a scout. The scout will randomly search for a new food source within the whole search space according to the following expression:

$$x_{ij} = x_j^{\min} + r\left(x_j^{\max} - x_j^{\min}\right), \tag{10}$$

where $x_j^{\min}$ and $x_j^{\max}$, respectively, represent the minimum and maximum of the $j$th dimension of all available food sources in the whole search space. $r$ is a random number between 0 and 1. The approach checks whether the maximum iterations number is reached. If yes, the PS-ABC approach ends and returns the global optimal solution. Otherwise, the approach begins to perform the next round of iterations. That is, all the employed bees begin neighbourhood search according to the expression (5). As described above, after a neighbourhood search if the fitness value $\eta_r$ of an employed bee's new candidate food source is not $D$, the employed bee can continue to have neighbourhood search and have $AL$ chances. If $AL$ times of neighbourhood search still cannot make it find a new food source with $\eta_r$ equal to $D$, its result of neighbourhood search is the best one of the $AL$ candidate food sources. If it is better than the old one, the food source position of each employed bee is updated to the result of neighbourhood search in the memory. Otherwise, the old one is kept. Then, the $\eta_r$ values of all employed bees are compared to get the largest ones. The fitness value $\delta P$ of each of the food sources with the largest $\eta_r$ values is calculated by its employed bee according to expressions (3) and (4). The $\delta P$ values of other food sources are set to 0. For onlooker bees' selection, the probability $p_i$ values of all food sources are calculated according to expressions (8) and (9). Each of the FSN onlooker bees reselects its employed bee to follow by using the probability wheel as shown in Figure 2.

After an onlooker bee reaches its employed bee's food source, it begins to perform neighborhood search according to expression (6). The new candidate food source $v_i$ of an onlooker bee is compared with the current food source $x_i$ of its employed bee in the memory. If $v_i \prec_{\eta_r,\delta P} x_i$, then the position $x_i$ of the food source is changed to be $v_i$; otherwise $x_i$ is kept as it is. Once all onlooker bees belonging to different employed bees finish their neighborhood search, the food sources of all the employed bees are compared according to $\prec_{\eta_r,\delta P}$ to obtain the global optimal food source of the second round of iterations. The second round of iterations ends. The third round of iterations will begin and so forth. At the end of each round of iterations, the PS-ABC approach needs to determine three problems. It firstly checks whether there exists any employed bee's food source which has not been improved continuously for LIMIT rounds of iterations and the food source is not the best one during LIMIT rounds of iterations. If yes, the employed bee has to abandon its current food source in the memory and become a scout to randomly search for a new food source according to expression (10). Then, the approach checks whether there exists any employed bee's food source which has not been improved continuously for LIMIT rounds of iterations and the food source is always the best one of all employed bees' food sources during LIMIT rounds of iterations. If yes, the approach stops iterative process and enters the next phase, which will be described in the following; if no, the PS-ABC approach will finally check whether the maximum number of iterations is reached or not. If it is met, the approach ends and returns the current global optimal food source (solution vector). Otherwise, the approach continues to the next round of iterations.

*4.4.4. Bayes Theorem Process of the PS-ABC Approach.* As mentioned above, after each round of iterations, once the algorithm finds that there exists any employed bee's food source which has not been improved continuously for LIMIT rounds of iterations as well as the food source is always the best one of all employed bees' food sources during LIMIT rounds of iterations, PS-ABC stops iterative search process and enters the PS-ABC approach's Bayes theorem process proposed in this paper. It is obvious that the whole search space can be seen as FSN search areas by FSN employed bees logically at this moment. Each area has an employed bee as a master and has zero or several onlooker bees to follow its employed bee as slaves to search together for food sources within this area as shown in Figure 3.

The solid points represent the employed bees and the hollow points represent the onlooker bees in Figure 3. Here we take use of Figure 3 to illustrate the Bayes theorem process of the PS-ABC approach. We can regard each round of iterations as a randomized trial $E$. We define the search space as a sample space $S$ of the randomized trial $E$. Let $A$ be a random event: $A$ = {a bee finds out the current global optimal food source (optimal solution vector)}. B1 (logical area 1), B2 (logical area 2), . . ., and B6 (logical area 6) are partitions of $S$. Assume that the PS-ABC has already finished $N$ rounds of iterations before stopping. Besides, assume that the number of rounds for which the food source of each employed bee has been a current global optimal food source within the $N$ rounds of iterations is $N_i$. In other words, $N_1 + N_2 + N_3 + N_4 + N_5 + N_6 = N$. So, we can get the following conditional probability value:

$$P(A \mid B_i) = \frac{N_i}{N}, \quad i \in \{1, 2, 3, 4, 5, 6\}. \tag{11}$$

It represents the probability with which the bees in logical area $i$ find out the current global optimal food source. The following probability value can also be obtained:

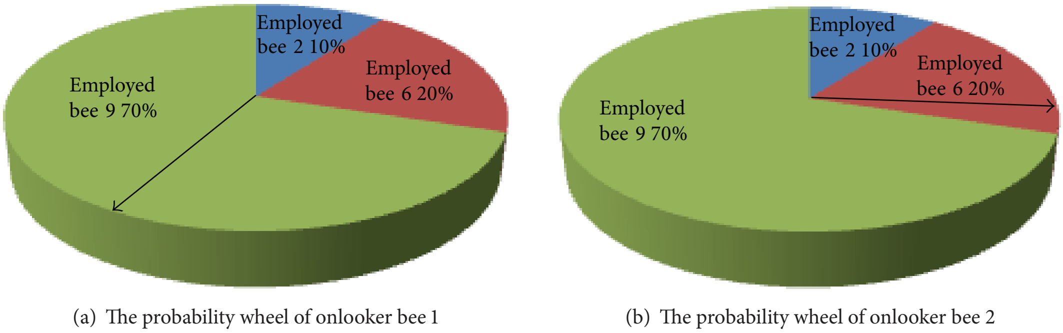
$$P(B_i) = \frac{\mathrm{Bee}_i}{\mathrm{Bee}}, \tag{12}$$

where $\mathrm{Bee}_i$ represents the number of bees in logical area $i$. Bee is denoted to the total number of bees. Its reasonability lies in that each bee has the same ability in searching for the food sources by exploring and exploiting the whole search space. Thus we can obtain $P(B_i)$ in expression (12) according to geometric probability. This is also the theoretical basis for partitioning the whole search space. In terms of the proposed problem and our definition, we can calculate $P(B_i \mid A)$ in order to make decisions in the next step according to the Bayes theorem as the following expression:

$$P(B_i \mid A) = \frac{P(B_i)\,P(A \mid B_i)}{\sum_{i=1}^{n} P(B_i) \cdot P(A \mid B_i)}, \quad i \in \{1, 2, 3, 4, 5, 6\}. \tag{13}$$

In these probability values obtained by calculating, the largest one $P(B_M \mid A)$ is picked out. It represents that the final optimal food source is most likely to be in logical area $M$. At this point, all bees perform neighborhood search around the food source which the employed bee $M$ is located at. The neighborhood search is repeatedly executed until the food source has not been improved for LIMIT rounds of neighborhood search continuously. At that moment, the PS-ABC approach returns the food source position. That is, the final solution vector is obtained. The algorithm ends.

Let us consider a problem in the proposed approach. We know that the solution vector of the proposed problem ought to be a sequence of integer numbers which denote the hosts in the resource pool. However, each food source of the PS-ABC is randomly initialized. What is more is that the expressions used by PS-ABC have coefficients which are limited between 0 and 1. Thus although the elements of each food source can be limited to an integer type in the implementation, the problem that the solution is not an integer still persists. To address this problem, we do not limit the elements to an integer type but employ the Smallest Position Value (SPV) rules presented in [13, 14]. In a nutshell, it is a rule which converts each food source's position vector given by the PS-ABC approach to a valid solution vector fit for the proposed problem. The process of applying the SPV rules into the proposed approach can be understood



(a) The probability wheel of onlooker bee 1

(b) The probability wheel of onlooker bee 2

Figure 2: Examples of the probability wheel for onlooker bees.

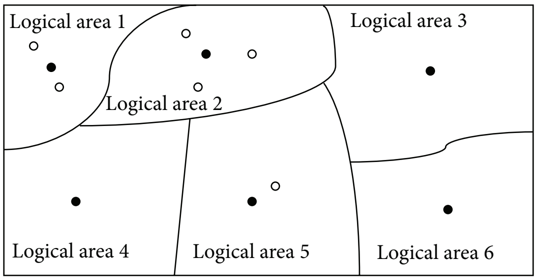


Figure 3: An example of logic area division of search space at some point.

as follows. First, the hosts in the resource pool should be numbered from 0 to $m-1$. Second, after the position of each food source is initialized randomly, all the elements of each position vector are sorted in ascending order and then are numbered from 0 to $n-1$ as well as having a modulo operation of $m$. For instance, in a time window $\Delta t$, there are four VMs to be migrated in a resource pool which has three hosts. If a position vector is $(-1.21, 3.29, -0.12, 1.26)$, it will be converted to $(0, 3, 1, 2)$ firstly and then to $(0, 0, 1, 2)$. At this point, the solution vector is useful and meaningful. It represents that the target host of VM 0 is host 0. The target host of VM 1 is host 0. The target host of VM 2 is host 1. The target host of VM 3 is host 2. In the PS-ABC, all the food source position vectors refer to the vectors which the original vectors have been converted to according to the SPV rules.

In the proposed PS-ABC approach based on ABC, the bees are divided into two categories of employed bees and onlooker bees. And after an employed bee abandons its food source, it will become a scout. We know that each food source has an employed bee, which can be regarded as the master of a search group, the core of a logical area and keeps the position information of the current food source of this logical search group in the memory. What is more is that it can recruit onlooker bees with a certain probability by sharing its fitness information. It is well-known that this kind of swarm intelligence approaches such as ABC, particle swarm optimization (PSO), and ant colony optimization (ACO) has some common characteristics (virtues and deficiencies). On one hand, all of them are heuristic approaches and suitable for dealing with large-scale problems as well as having abilities in parallel or concurrent computing. On the other hand, they have some deficiencies due to their own characteristics such as premature and local-best solution. In PS-ABC, these onlooker bees as slaves are the main force of searching while the employed bees as masters should guarantee that the search focuses are rational and uniform.

In order to fully exert the virtues of ABC and eliminate its deficiencies, two important measures are adopted in the proposed PS-ABC approach. In the ABC approach, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. However, at the beginning of PS-ABC, the initialization of food sources (i.e., the initial positions of employed bees) is also crucial for the next global search. The random initialization is not employed but the uniform random initialization method is presented and employed. It makes the PS-ABC approach have a better global exploration potential. It will get double results with half the effort. We know that in a robust search process, exploration and exploitation processes must be carried out together. Obviously, the core competency of ABC idea is exactly as follows: it can more easily make exploration and exploitation processes be carried out together by the cooperation of employed bees, onlooker bees, and scouts. However, its efficiency is not high enough since the classical ABC approach employs the roulette method, which does not take selective pressure into consideration, for onlooker bees' selection. In an iterative search algorithm, it should have better global search ability in the early iterations and it should have better local search and convergence ability in the later iterations. Thus, the selective pressure should change with the iterative process being keept on. More specifically, the onlooker bees should not be greedy and go to all employed bees as uniformly as possible in the early iterations in order to make the whole search space explored better and thus to avoid being trapped into local optimal food sources. In the later iterations, the onlooker bees gradually become greedy. That is, the employed bee with a better food source should have a larger probability value for onlooker bees' selection as well as to make the approach have high-efficient convergence ability and avoid premature. To achieve this goal, we have thought of simulated annealing (SA) idea [15, 16]. Further, by studying the SA idea, we have found the Boltzmann selection policy and introduced it into the proposed PS-ABC approach to thus achieve the self-adaptive selective pressure adjustment method. It makes the global exploration and local exploitation of employed bees, onlooker bees, and scouts more high-efficient and meaningful.

Starting from this intuition, the PS-ABC approach which aims at live VM migration policy gives consideration to the power saving optimization to achieve a more efficient cloud data center by designing and utilizing the ABC. During the course of the study, we have found that the search process of PS-ABC can be regarded as the repetitive randomized trials since it has the characters of randomized trials such as the nondeterminacy of results, the repeatability of trials, and the diversity of results and so forth. This lies in the iterative process of PS-ABC and its expressions with the random parameters. What is more is that in the PS-ABC approach, the whole search space can be divided into several logical search areas like some partitions of a sample space. This is peculiar to the ABC-based approach due to its design mode and search pattern. That is, an employed bee with some onlooker bees represents a logical search area. Its basis is that each bee has the same abilities in exploring and exploiting the search space. So, the whole search space can exactly be divided into several logical partitions (i.e., their union is the whole search space and their intersection is the empty set). At this point, the search of the global optimal food source can be seen as a random event in the sample space (search space).

Further, under this underground we can utilize the existing conditions and Bayes theorem to get the information about which logical area is most likely to have the global optimal solution. Upon obtaining this information, all bees can emphatically search this area and thus to quickly make the approach converge to the global optimal food source (solution vector). The triggering condition of Bayes theorem in PS-ABC has been described above. As a matter of fact, the design of its triggering condition has considered two aspects of problems at the same time. One is that if a food source has not been improved continuously for too many (LIMIT) rounds of iterations and it is always the optimal food source during these rounds of iterations, the food source will probably be the final global optimal solution or the global optimal solution is near it. However, we need to verify it. The core goal and idea of the Bayes theorem process is right here. Based on it, the triggering condition is set. The other is that the classical ABC approach has only considered the situation that a food source has not been improved continuously for too many (LIMIT) rounds of iterations and it is not the global optimal food source in each of these rounds of iterations. At that time, that food source will be abandoned by its employed bee. However, if the above food source is always the global optimal during these rounds of iterations, the classical ABC does not do any response. It is obvious that at this moment the algorithm should take action to respond to this phenomenon and thus to optimize the whole process. In addition, this is also the reason for that the round number of triggering condition is also set to LIMIT.

There are many parameters in the proposed PS-ABC approach. Most of them have great influences on PS-ABC. The maximum number MNI of iterations and the number FSN of employed bees (or onlooker bees) are two important parameters and their values are related to the efficiency and accuracy of PS-ABC. For instance, if the bees number FSN increases, the computation time (convergence time) of PS-ABC will increase and the possibility that the approach is converged to the global optimal solution will also increase. That is, the global search ability will strengthen. Moreover, the required iteration number MNI will decrease with FSN increasing. The maximum number MNI of iterations should be neither too small nor too large. Evidently, the size of the MNI value should make the current global optimal food source of each round of iterations of the PS-ABC approach exactly converge to the final global optimal food source as much as possible. Therefore, both the MNI value and the FSN value are the experience problems and need to perform lots of experiments to obtain fit values. In the proposed PS-ABC approach, FSN is set to 20 and MNI is set to 100 based on a large number of experiments.

Also, the controlling parameter LIMIT is another important value in PS-ABC since it has a direct impact on the implementation of the whole PS-ABC approach. It controls the transformation from an employed bee to a scout and controls when to enter the Bayes theorem process. Both of the two steps are crucial for PS-ABC. It is difficult for PS-ABC to assign to the controlling parameter LIMIT a fit value as it is related to several factors. Thus, it is an open problem which needs be studied further. In this paper, the LIMIT value is set to 20. Similarly, the $AL$ value is an important controlling parameter proposed in the PS-ABC approach. Its value is an open problem. In this paper, it is set to 10. Besides, the initial temperature $T$ and its attenuation factor $\alpha$ of Boltzmann selection policy are also important parameters. $T$ should be initialized to an enough big value and it makes the corresponding algorithm have a stronger global search in the early iteration. Then $T$ is gradually decreased with the iterative number increasing, which makes the search process gradually converge to an optimal solution in the later iteration. Thus, the values of $T$ and $\alpha$ are important for Boltzmann selection idea. They should be set to the appropriate values as much as possible.

## 5. Evaluation

In this section, we have experimentally verified the proposed PS-ABC approach. The experiments have included evaluating the effect of power saving of PS-ABC and verifying the migration performance of PS-ABC migration policy on the failure rate of migration events. Besides, two auxiliary experiments have been conducted to evaluate a related parameter and get a better power management policy. In order to simulate a dynamic cloud data center, we have utilized an event driven simulator named CloudSim toolkit [17]. The CloudSim framework enables the accounting of the total power consumed by the system during the simulation period. In one word, the calculating of power consumption has been achieved by the CloudSim platform, which has provided a class including the methods getPower( ). CloudSim allows such simulation scenarios by supporting dynamic creation of different kinds of entities and can add and remove data center entities at run-time. This functionality has achieved simulating dynamic cloud environment where system components can join, fail, or leave the system randomly [17]. On CloudSim platform, we compare the proposed PS-ABC approach with the random migration policy and the optimal migration policy based on Genetic Algorithm (GA) by power consumption and the number of invalid VM migration. We have prepared four different kinds of experiments to evaluate and test the proposed approach. The results demonstrate that the proposed approach not only has a better power saving but also has a better migration performance compared to random migration policy and GA migration policy. Especially for a large number of migration events, the PS-ABC approach shows the stability and the better performance.

*5.1. Experimental Scenarios.* On CloudSim platform, a resource pool consisting of 100 hosts is created. These hosts have varying computing resource. 24 batches of virtual machine migration requests containing 13 requests randomly belonging to different hosts and with different resource requirements are created. The proposed PS-ABC module is invoked and fetches the resource information and state of the cloud resource pool periodically.

*5.2. Comparison of GA, Random-Migration, and PS-ABC in Power Consumption.* The experiment is designed for verifying the efficiency and availability of PS-ABC in power saving due to the placement selection of live VM migration during the operation of a cloud data center. In this scenario, we compare PS-ABC with Random-Migration and GA optimal migration policy by power consumption during the operation of the simulated cloud data center. In this experiment, $\Delta t$ is set as 600 seconds and the migration events of each batch are uniformly distributed within an hour, which includes $6\Delta t$s. Besides, the host's resource change rate is set as 1 time per half an hour.

As illustrated in Figure 4, at the end of each week, the cloud data center implementing PS-ABC consumes the least power than the cloud data center implementing GA and Random-Migration. The cloud data center implementing GA has a less power consumption than the cloud data center implementing Random-Migration. Moreover, it can also be seen that the cloud data center implementing PS-ABC has the least incremental power consumption between two weeks in view of the general trend. However, the incremental power consumption of Random-Migration has been increasing gradually between two weeks. This is because the Random-Migration policy is not sensitive enough to changes of the cloud environment and thus may cause more migration events or selects randomly an unreasonable target host in power consumption; with the passage of time, this phenomenon will be more and more obvious. For the GA policy, although it has a relatively less power consumption than Random-Migration, it is not so excellent as PS-ABC. It is well-known that GA is a widely used classical heuristic intelligent algorithm. However, compared with PS-ABC, it has some shortages. GA needs to code and decode each individual to represent a specific solution and enable the crossover and mutation operators. GA is not suitable for dealing with large-scale problems and it is easy to premature since its degree of parallelization and performance is not high enough. But for PS-ABC, it does not need the coding and decoding as well as genetic operators. From its algorithm and process, it can be seen that PS-ABC's degree of parallelization and performance is higher than that of GA. Accordingly, PS-ABC has a better power saving efficiency than GA naturally.

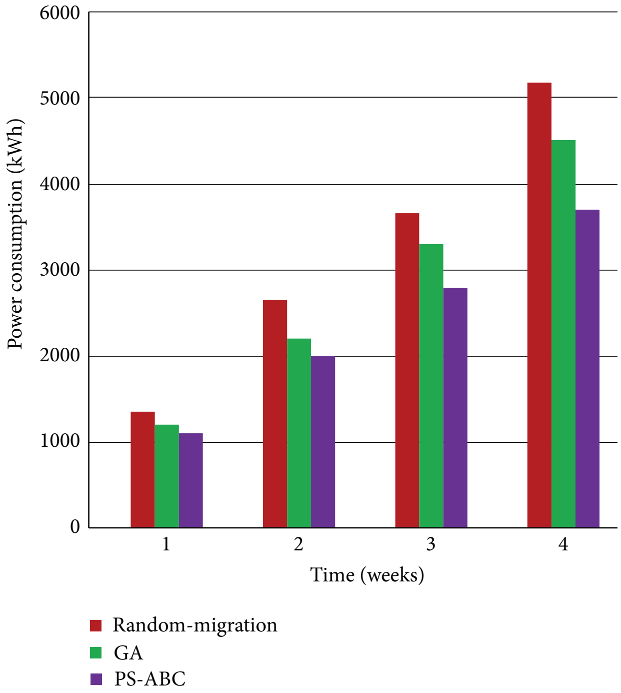


Figure 4: Comparison of GA, Random-Migration, and PS-ABC in power consumption.

*5.3. Comparison of the Number of Failures in VM Migration Events.* In the experiment scenario, the dynamic host failure is simulated by CloudSim by scheduling some host failure events and host shut down events to occur during the placement selection interval. These events have caused some failures in VM migration due to the nonavailability of the selected hosts for some VM migration requests. As illustrated in Figure 5, we have compared the PS-ABC approach with random migration and optimal migration implemented by GA. In the simulated cloud data center, with the increase of the number of VM migration requests, the cloud data center implementing Random-Migration and GA result in more number of invalid VM migrations, whereas the PS-ABC performs better in finding the fit hosts in the dynamic resource pool. This is because the memory data is outdated in the GA and Random-Migration. They cannot have an adjustment with the environment changed. Conversely, the PS-ABC policy which utilizes the Boltzmann selection policy and Bayes theorem is quite efficient in responding to the host failures during the interval as well as having a fit adjustment in a better manner by searching out the new available hosts that can meet the resource requirements of the VM migration requests. As described above, the proposed PS-ABC approach

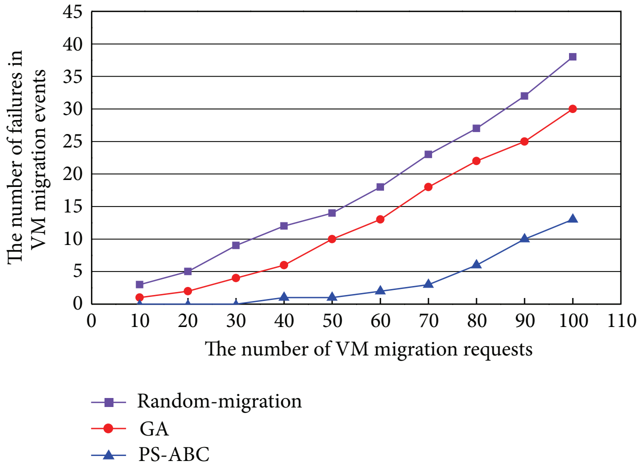


Figure 5: Comparison of the number of failures in VM migration events.

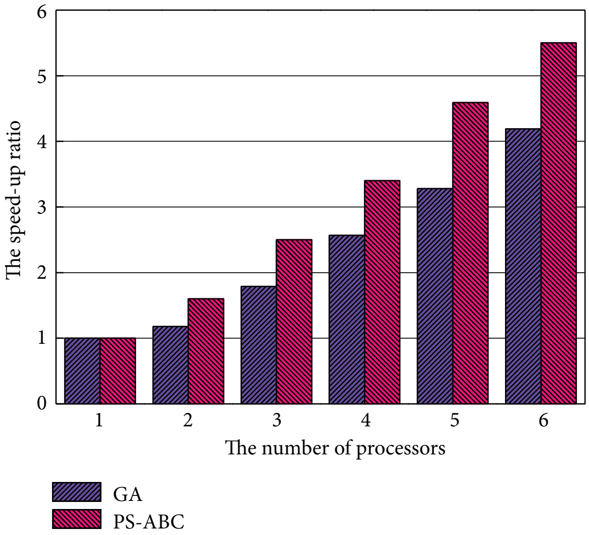


Figure 6: Comparison of GA and PS-ABC in speed-up ratio.

is designed to primarily meet performance requirements. On this basis, the power consumption is decreased as much as possible.

*5.4. Comparison of GA and PS-ABC in Speed-Up Ratio.* In this experimental scenario, we will evaluate the speed-up ratios of GA and PS-ABC with the number of processors increasing. The speed-up ratio $S_p$ can be expressed as follows:

$$S_p = \frac{T_1}{T_p}, \tag{14}$$

where $T_1$ is the processing time of a task under the condition of single processor and $T_p$ refers to the running time of the same task under the condition of $p$ processors. Thus, the speed-up ratio $S_p$ can be used to measure the performance and effect of program parallelization. As shown in Figure 6, the experiment shows the speed-up ratios of PS-ABC and GA under these conditions of different number of processors. With the increase of the number of processors, the speed-up ratio of PS-ABC is always larger than that of GA. The proposed PS-ABC approach is more close to the linear speed-up. For the PS-ABC approach, it can see the whole search space as multiple independent search subspace logically. Each of these subspaces performs its own search independently and then the employed bees, as the representatives of these logic search subspaces, share their information uniformly. The mechanism and process of PS-ABC is easy to parallel processing. Whereas for GA, it does not have these features and advantages. Its mechanism is suitable for centralized processing relatively. Thus, the performance and efficiency of PS-ABC are more excellent than those of GA. In particular, in cloud data centers, the PS-ABC approach is faster and more efficient.

*5.5. Comparison of the Incremental Power Consumption in a Cloud Data Center with Varying Percentage of Load.* In this experimental scenario, we compare the incremental power consumption of GA and PS-ABC in a cloud data center with varying percentage of load. The load mentioned refers to the load of the whole cloud data center and is not the load of some physical host. The varying percentage of load means that this experiment has been done in the same cloud data center with a different load. As illustrated in Figure 7, with the increase of the percentage of load in the cloud data center, the incremental power consumption of PS-ABC is less than that of GA. We know that, if the load in the hosts is heavier, the scale of the problem to be solved will be larger. That is, after a time window $\Delta t$, the number of VMs which need to be migrated is larger. Compared with GA, the PS-ABC approach is more suitable for solving the relatively larger-scale problems. Its final returned optimal solution vector is more accurate. Therefore, PS-ABC is more efficient power-saving migration policy from a long-term view for the cloud environment with a heavy load.

*5.6. Trade-Off between Power Saving and Performance Fulfillment.* The experimental scenario is conducted to find the optimal power management policy which balances the benefits due to power saving and performance fulfillment. Since live VM migration events are time critical VM requests and the cloud service provider should meet strict Service Level Agreements (SLA) compliance, any violations in SLA in terms of performance loss of VMs will result in penalty cost on the provider of a cloud data center. The performance loss mentioned may be caused by power saving, live VM migration, network bandwidth, throughput, and so forth. However, this paper has not focused on these problems. This experiment is designed to only find out a better power management policy suitable for the cloud environment implementing the proposed PS-ABC to have a trade-off between the benefit of power saving and penalty cost due to SLA violation. There are four different policies to be formulated. The first policy is On/Off policy, wherein all idle hosts are switched off. It can be seen that the policy gives the best power saving, but it causes the high penalty cost obviously. The single-DSS policy is

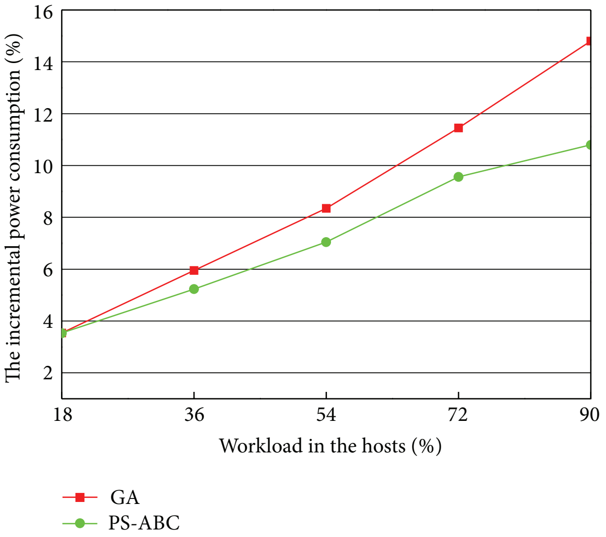


Figure 7: Comparison of the incremental power consumption in the cloud data center with varying percentage of load.

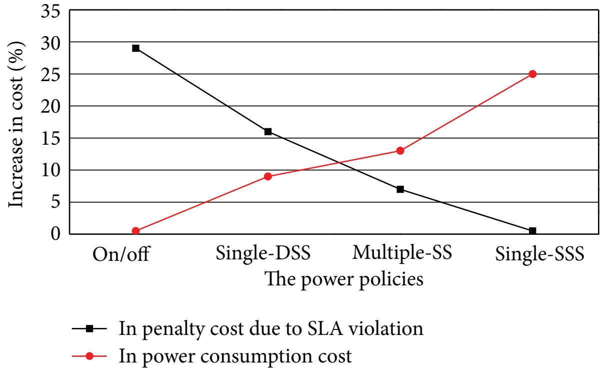


Figure 8: The trade-off between power and performance with different power management policies.

the second policy, wherein all idle hosts are switched to deep sleep state. It results in an increase in the power consumption cost, but the penalty cost is reduced significantly. The third policy is single-SSS, wherein all idle hosts are switched to shallow sleep state. There is no penalty cost as SLA violation is absent in this policy. However, the enormous increase in the power consumption cost is caused. The multiple-SS is the fourth policy, wherein some of the idle hosts are kept in deep sleep and others are kept in shadow sleep state according to a short-term prediction technology. From its meaning and as shown in Figure 8, it can be known that the multiple-SS policy gives the optimal cost trade-off relatively.

## 6. Conclusion and Future Work

In this paper, a novel placement selection policy of live VM migration PS-ABC is proposed and we give its algorithm, implementation, and evaluation. It is based on the improved ABC-based approach which employs the Boltzmann selection idea and Bayes theorem. In the improved ABC-based approach, we introduce the binary search idea into the classic ABC approach to achieve the uniform random initialization and thus to make the whole PS-ABC approach have a better global search potential and capacity at the very beginning. In the proposed PS-ABC approach based on ABC approach, we have employed *AL Times of Attempts* policy of neighbourhood search in order to improve the efficiency of searching without too much additional overhead while making the first fitness function of performance requirements become a constraint condition and thus to be consistent with the real needs. Also, PS-ABC does not employ the roulette wheel selection for onlooker bees' selection but utilizes the Boltzmann selection policy of simulated annealing idea to make the proposed approach have the self-adaptive selective pressure adjustment and thus to achieve having the better global exploration ability in the early stage and the better exploitation ability in the later stage. What is more is that we have introduced the Bayes theorem into the proposed PS-ABC approach. A sample space and its all elements are presented and formalized. By calculating probability values and using them, the PS-ABC approach can be faster converged to the global optimal solution vector. It not only makes the PS-ABC approach have abilities in fast convergence in some cases but also fills in the gap of the classical ABC approach. It has shown that the proposed PS-ABC approach has the strong mathematical basis.

PS-ABC achieves the high-efficiency of power consumption and the stability of requirement performance. It minimizes not only the incremental power consumption of a cloud data center but also the number of failure in VM migration events relatively. It aims at achieving the better power conservation during the long-term operation of a cloud data center and protects the performance of VM running. In the proposed PS-ABC approach, there are some open problems which need to be studied further and experience problems which need many experiments to gradually get a better solution. The maximum number MNI of iterations and the number FSN of employed bees (or onlooker bees) as well as the $T$ value of Boltzmann selection policy are the experience problems and we need to perform several experiments to obtain the fit values to thus make the PS-ABC approach efficient and feasible. The controlling parameters LIMIT and $AL$ of our approach as well as the attenuation factor $\alpha$ value of Boltzmann selection policy are the open problems which need to be further widely researched. In this paper, all the parameters are set to the fit values, respectively.

To evaluate the PS-ABC approach, we have conducted several experiments on the CloudSim platform. Firstly, in the comparison experiment of GA, Random-Migration, and PS-ABC in power consumption, the PS-ABC approach has the least incremental power consumption during the long-term operation of the cloud data center. Secondly, in the comparison experiment of the number of failures in VM migration events, the PS-ABC approach has the least number of failures in VM migration events than that of random migration and GA optimal migration. Thirdly, in the comparison experiment of GA and PS-ABC in Speed-up Ratio, the result has shown that the proposed PS-ABC approach has the better speed-up ratio and correspondingly it has the better efficiency compared with the GA migration policy. Finally, in the comparison experiment of the incremental power consumption in the cloud data center with varying percentage of load, the result has shown that the PS-ABC approach is more efficient in power saving than random migration and GA optimal migration for the cloud data center with a heavy load during the long-term operation of a cloud data center. In order to balance the relationship between the power conservation and performance fulfillment, we have also performed an experiment to find out the better power management policy and proved that the multiple-SS policy is the best selection. The final experimental results show that PS-ABC is an effective placement selection policy of live VM migration for power conservation.

Aiming to further improve the performance of PS-ABC, we plan to study the robustness of PS-ABC in the next step work. PS-ABC should have abilities in dealing with some sudden matters and be combined with the mechanism of live VM migration for achieving a more efficient hybrid management. Besides, in this paper, PS-ABC is used for the placement selection of live VM migration in LAN. In the future, we will extend PS-ABC in WAN. In the next work and experiments, the experience problems and the open problems presented in this paper will also be researched further. If possible, we will research and implement the proposed PS-ABC approach in a real cloud computing environment as well as evaluating its performance and efficiency.

## Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their valuable comments. The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

[1] P. Barham, B. Dragovic, K. Fraser et al., "Xen and the art of virtualization," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, pp. 164–177, October 2003.

[2] Y. Li, W. Li, and C. Jiang, "A survey of virtual machine system: Current technology and future trends," in *Proceedings of the 3rd International Symposium on Electronic Commerce and Security (ISECS '10)*, pp. 332–336, July 2010.

[3] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[4] C. Rusu, A. Ferreira, C. Scordino, A. Watson, R. Melhem, and D. Mossé, "Energy-efficient real-time heterogeneous server clusters," in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 418–427, San Jose, Calif, USA, April 2006.

[5] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," *Cluster Computing*, vol. 12, pp. 1–15, 2009.

[6] A. Verma, P. Ahuja, and A. Neogi, "PMapper: power and migration cost aware application placement in virtualized systems," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, pp. 243–264, Spinger, New York, NY, USA, 2008.

[7] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "EnaCloud: An energy-saving application live placement approach for cloud computing environments," in *Proceedings of the IEEE International Conference on Cloud Computing (CLOUD '09)*, pp. 17–24, September 2009.

[8] R. Jeyarani, N. Nagaveni, and R. Vasanth Ram, "Self adaptive particle swarm optimization for efficient virtual machine provisioning in cloud," *International Journal of Intelligent Information Technologies*, vol. 7, no. 2, pp. 25–44, 2011.

[9] R. Jeyarani, R. Vasanth Ram, and N. Nagaveni, "Implementation of efficient light weight internal scheduler for high throughput grid environment," in *Proceedings of the National Conference on Advanced Computing in Computer Applications (NCACCA '09)*, E. Jeyakumar and R. Rangarajan, Eds., pp. 283–289, Coimbatore, India, 2009.

[10] S. Sadhasivam, R. Jayarani, N. Nagaveni, and R. Vasanth Ram, "Design and implementation of an efficient two-level scheduler for cloud computing environment," in *Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom '09)*, pp. 884–886, Kottayam, India, October 2009.

[11] 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.

[12] 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.

[13] M. F. Tasgetiren, M. Sevkli, Y.-C. Liang, and G. Gencyilmaz, "Particle swarm optimization algorithm for permutation flowshop sequencing problem," in *Ant Colony Optimization and Swarm Intelligence*, vol. 3172 of *Lecture Notes in Computer Science*, pp. 382–389, Springer, Heidelberg, Germany, 2004.

[14] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz, "Particle swarm optimization algorithm for single machine total weighted tardiness problem," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 1412–1419, Portland, Ore, USA, June 2004.

[15] T. Mahnig and H. Mühlenbein, "A new adaptive Boltzmann selection schedule SDS," in *Proceedings of the Congress on Evolutionary Computation*, pp. 183–190, May 2001.

[16] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 869658, 9 pages
http://dx.doi.org/10.1155/2013/869658

*Research Article*

# A New Collaborative Recommendation Approach Based on Users Clustering Using Artificial Bee Colony Algorithm

**Chunhua Ju[1,2] and Chonghuan Xu[1,3]**

[1] *Center for Studies of Modern Business, Zhejiang Gongshang University, Hangzhou 310018, China*
[2] *College of Computer Science & Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*
[3] *College of Business Administration, Zhejiang Gongshang University, Hangzhou 310018, China*

Correspondence should be addressed to Chonghuan Xu; talentxch@gmail.com

Received 5 September 2013; Accepted 17 October 2013

Academic Editors: T. Chen and J. Yang

Although there are many good collaborative recommendation methods, it is still a challenge to increase the accuracy and diversity of these methods to fulfill users' preferences. In this paper, we propose a novel collaborative filtering recommendation approach based on $K$-means clustering algorithm. In the process of clustering, we use artificial bee colony (ABC) algorithm to overcome the local optimal problem caused by $K$-means. After that we adopt the modified cosine similarity to compute the similarity between users in the same clusters. Finally, we generate recommendation results for the corresponding target users. Detailed numerical analysis on a benchmark dataset *MovieLens* and a real-world dataset indicates that our new collaborative filtering approach based on users clustering algorithm outperforms many other recommendation methods.

## 1. Introduction

Nowadays, the Internet continues growing at an exponential rate in its size and complexity. For the users of numerous web sites it becomes increasingly difficult and time consuming to find the information they are actually looking for. As a consequence, how to efficiently help users filter out the unwanted information and find what is really useful for them is a challenging problem for information filtering. Recommender systems have proven to be an effective method to deal with the problem of information overload in finding interesting products. They not only help decide which products to offer to an individual customer but also increase cross-sell by suggesting additional products to the customers and improve consumer loyalty because consumers tend to return to the sites that better serve their needs. With the development of recommender systems, various kinds of recommendation techniques have been proposed, including collaborative filtering (CF) [1–3], content-based filtering [4, 5], $K$-nearest neighbor ($K$-NN) [6–8], diffusion approach [9, 10], and spectral analysis [11, 12]. A collaborative filtering approach builds a model to predict what users will like according to their similarity to other users based on collecting and analyzing a large amount of information on users' behaviors, activities, or preferences. Content-based filtering methods are based on information about and characteristics of the products that are going to be recommended. They try to recommend products that are similar to those that a user liked in the past. $K$-NN is one of the most fundamental and simple classification methods for classifying objects based on the properties of their closest neighbors in the feature space. In $K$-NN, an object is classified through a majority vote of its neighbors, with the object being assigned to the class most common amongst its $k$ nearest neighbors. Diffusion approach based on the mass diffusion principle generates the recommendation results for target users on a user-object bipartite network. Spectral analysis is a new recommendation algorithm that relies on the singular value decomposition (SVD) of the rating matrix.

Otherwise, more and more scholars apply clustering or classification techniques into recommendation methods in order to enhance the recommending accuracy. We know that users in a cluster will have similar interests; thus, if a product is selected by these users it will be suitable to the target user. It leads to more accurate recommendations. So, in this paper, we propose a novel collaborative filtering recommendation approach based on $K$-means clustering algorithm. The reason why we choose $K$-means is that it is the most popular class of clustering algorithms while simple and fast. However, $K$-means algorithm highly depends on the initial states and always converges to the nearest local optimum from the starting position of the search. Therefore, in the process of clustering, we use artificial bee colony (ABC) algorithm to overcome these problems. And then we adopt the modified cosine similarity considering products' popularity degrees and users' preference degrees to compute the similarity between users in the same clusters. Finally, we generate the recommendation results for target users. Detailed numerical analysis on a benchmark dataset *MovieLens* and a real-world dataset indicates that our new collaborative filtering approach based on users clustering algorithm outperforms many other methods.

The main contribution of this paper is summarized as follows: we propose a novel collaborative filtering recommendation approach to generate good recommendation results for target users. We import $K$-means clustering algorithm into it in order to enhance the accuracy of recommendations. And we modify the standard cosine similarity for the sake of more accuracy and diversity of the recommendation results. The remainder of this paper is organized as follows. In Section 2, a review of related work is given. In Section 3, our recommendation model and new collaborative filtering method based on users clustering using artificial bee colony algorithm are described. Section 4 provides experimental results and analysis of the proposed method on a benchmark dataset and a real-world dataset. Finally, we draw conclusions in Section 5.

## 2. Related Work

In this section, we will introduce some good collaborative recommendation techniques and then describe certain relevant conceptions.

Liu et al. [13] proposed a novel method to compute the similarity between congeneric nodes on bipartite network. They considered the influence of a node's degree and then presented a modified collaborative filtering (MCF) to substitute the standard cosine similarity. Kim et al. [14] proposed a collaborative approach to user modeling for enhancing personalized recommendations to users. Their approach first discovered some useful and meaningful user patterns and then enriched the personal model with collaboration from other similar users. López-Nores et al. [15] presented a new strategy called property-based collaborative filtering (PBCF) to address problems of recommender systems by introducing a new filtering strategy, centered on the properties that characterized the items and the users. Tsai and Hung [16] assessed the applicability of cluster ensembles to collaborative filtering recommendation. They used two well-known clustering techniques and three ensemble methods. The experimental results based on the *MovieLens* dataset showed that cluster ensembles could provide better recommendation performance than a single clustering technique in terms of recommendation accuracy and precision. Altingovde et al. [17] explored an individualistic strategy which initially clustered the users and then exploited the members within clusters, but not just the cluster representatives, during the recommendation generation stage. They provided an efficient implementation of this strategy by adapting a specifically tailored cluster-skipping inverted index structure. Wu et al. [18] presented a novel modified collaborative recommendation method called div-clustering to cluster Web entities in which the properties were specified formally in a recommendation framework, with the reusability of the user modeling component considered. Choi and Suh [19] proposed a new similarity function in order to select different neighbors for each different target item. In the new similarity function, the rating of a user on an item was weighted by the item similarity between the item and the target item. Gan and Jiang [20] proposed a network-based collaborative filtering approach to overcome the adverse influence of popular objects while achieving a reasonable balance between the accuracy and the diversity. Their method started with the construction of a user similarity network from historical data by using a nearest neighbor approach. Based on this network, they calculated discriminant scores for candidate objects and further sorted the objects in nonascending order to obtain the final ranking list. Bilge and Polat [21] proposed a novel privacy-preserving collaborative filtering scheme based on bisecting $K$-means clustering in which they applied two preprocessing methods. The first preprocessing scheme dealt with scalability problem by constructing a binary decision tree through a bisecting $K$-means clustering approach while the second produced clones of users by inserting pseudo-self-predictions into original user profiles to boost accuracy of scalability-enhanced structure.

*2.1. K-Means Algorithm.* $K$-means is a rather simple but well-known algorithm for grouping objects. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of $n$ objects. Each object can be thought of as being represented by some feature vector in a $p$-dimensional space. This algorithm starts by guessing $k$ cluster centers and then iterates the following steps until convergence is achieved [22].

(1) Clusters are built by assigning each element to the closest cluster center.

(2) Each cluster center is replaced by the mean of the elements belonging to that cluster. The algorithm is described as follows.

Set dataset $X = \{x_i\}$, $i = 1, 2, 3, \ldots, n$, and assemble them as $k$ cluster, that is, to divide the dataset as the follows:

$$X = \cup_{j=1}^{k} C_j, \qquad C_{j1} \cap C_{j2} = \emptyset, \quad j1 \neq j2. \tag{1}$$

$C_j$ is the arbitrary cluster.

*Step 1.* Initialize cluster centers $c_j$, $j = \{1, \ldots, k\}$.

*Step 2.* Assemble the dataset $X$ by cluster centers, as

$$c_j := \left\{ x \left| \left\| x - c_j \right\| = \min \right| , x \in X \right\}, \quad j = 1, 2, \ldots, k. \tag{2}$$

In which $\| * \|$ as some kind of norm, as clustering often be processed in Euclidean space in fact, the norm mentioned above often be set as 2-norm. $c_j$ is the center of the cluster $C_j$.

*Step 3.* Update cluster centers as follows:

$$c_j := \left(\frac{1}{K_j}\right)\sum x \in C_j, \quad j = 1, 2, \ldots, k, \tag{3}$$

where $K_j$ is the number of data in cluster $C_j$.

*Step 4.* Stop. If the cluster centers do not change or the clustering converge towards some kind of value, the iteration is stop. For instance, the clustering converge condition meets a cost function $f_i$:

$$f_i = \frac{1}{k}\sum_{j=1}^{k}\sum_{x_i \in C_j} d\left(x_i, c_j\right). \tag{4}$$

Though the users similarity computation, we can find some products selected by the test users who have much similarity to the target user.

*2.2. Artificial Bee Colony Algorithm.* Artificial bee colony (ABC) is one of the most recently defined algorithms by Karaboga [23] in 2005, motivated by the intelligent behavior of honey bees. It is a very simple, robust, and population-based stochastic optimization algorithm. The performance of the ABC algorithm is compared with those of other well-known modern heuristic algorithms such as differential evolution (DE) and particle swarm optimization (PSO) on constrained and unconstrained problems [24–26]. In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers, and scouts. A food source represents a possible solution to the problem to be optimized. The nectar amount of a food source corresponds to the quality of the solution represented by that food source. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source has been abandoned by the bees becomes a scout.

The main steps of the algorithm can be described as follows.

*Step 1.* Generate the initial population of solutions by using a random approach. Let $X_i^j$ represent the $i$th feasible solution (food source). Each feasible solution is generated as follows:

$$X_i^j = X_{\min}^j + \text{rand}\,(0, 1)\left(X_{\max}^j - X_{\min}^j\right), \tag{5}$$

where $X_{\max}^j$ and $X_{\min}^j$ are the lower and upper bounds for the dimension $j$, respectively.

*Step 2.* Produce new solutions for the employed bees, evaluate them, and apply the greedy selection process. The formula of the selection can be expressed as

$$V_i^j = X_i^j + \text{rand}\,[-1, 1]\left(X_i^j - X_k^j\right), \tag{6}$$

where $j \in \{1, 2, \ldots, D\}$, $k \in \{1, 2, \ldots, \text{SN}\}$, and $k \neq i$. $D$ is the dimension of the problem, representing the number of parameters to be optimized. SN is the number of the food sources and equals the number of employed bees or onlooker bees.

*Step 3.* Calculate the probabilities of the current sources with which they are preferred by the onlookers.

*Step 4.* Assign onlooker bees to employed bees according to probabilities, produce new solutions, and apply the greedy selection process. Formula (7) is used to calculate the probability value used by the onlooker bees for discovering promising regions in the search space:

$$p_t = \frac{\text{fit}_t}{\sum_{j=1}^{\text{SN}} \text{fit}_j}, \tag{7}$$

where $\text{fit}_t = 1/(1 + f_i)$ and $f_i$ is the objective function.

*Step 5.* If the search times *Bas* of an employed bee is more than threshold *limit*, stop the exploitation process of the sources abandoned by bees and send the scouts in the search area for discovering new food sources, randomly.

*Step 6.* Memorize the best food source found so far.

*Step 7.* If the termination condition is not satisfied, go to Step 2; otherwise, stop the algorithm.

## 3. Collaborative Recommendation Approach Based on Users Clustering

In this section, we describe the proposed collaborative filtering recommendation approach based on $K$-means clustering algorithm. In the process of clustering, we use ABC algorithm to overcome the local optimal problem of the $K$-means clustering algorithm. After that we adopt the modified cosine similarity which considers products' popularity degrees and users' preference degrees to compute the similarity between users in the same clusters. Finally, we generate a recommendation list made up of these products (objects) and then recommend them to the target users. Figure 1 shows the framework of our proposed recommendation model.

There are three phases in this framework.

*(1) User Clustering.* In order to enhance the accuracy of recommendation results, we use $K$-means clustering method to cluster users before recommending. As we know that users in a cluster will have similar interests, thus, if a product is selected by these users, it will be suitable to the target user. It leads to more accurate recommendations. For the sake of overcoming the local optimal problem of this clustering method, we bring in ABC algorithm. The steps are shown in Section 3.1.

*(2) Similarity Computation.* Firstly, we give some definitions. We assume that there is a recommendation model which consists of $m$ users and $n$ objects, and each user has selected some objects. The relationship between users

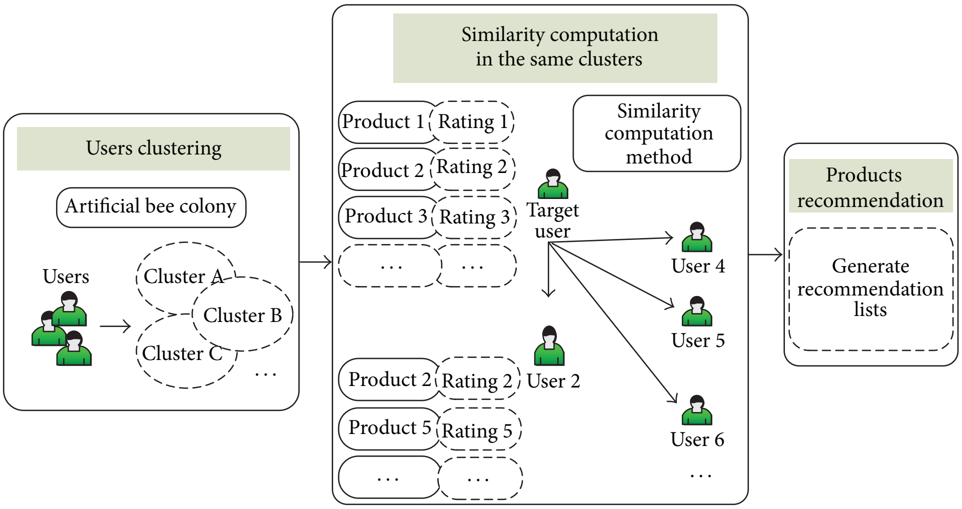


Figure 1: The framework of recommendation model based on users clustering.

and objects can be described by a bipartite network. Let $U = \{u_1, u_2, \ldots, u_m\}$ denote users set and $O = \{o_1, o_2, \ldots, o_n\}$ denote objects set, and the recommendation model can be fully described by an $m \times n$ adjacency matrix $A = \{a_{ij}\}$, where $a_{ij} = 1$ when object $j$ is selected by user $i$; otherwise, $a_{ij} = 0$. After that we use modified collaborative filtering method to compute the similarity between users. The detailed process of this computation is described in Section 3.2.

*(3) Products Recommendation.* In the previous phase, we compute the similarity between target user and others based on the influence of each object node including popular degree and preference degree. In this step, we calculate the value of comprehensive preference degree of each product unselected by the target user. Afterwards, the products with high comprehensive preference degree are used to compile a recommendation list in descending order. At last we recommend top $L$ products to the target user. In general, the number $L$ is no more than 100.

*3.1. K-Means Clustering Method Using ABC Algorithm.* In the process of users clustering, we use ABC algorithm to overcome the local optimal problem of $K$-means. As we know, in traditional $K$-means algorithm, users need to preset the $k$ value and centre points which will often have a big influence on cluster results. In our clustering method, we use ABC algorithm to determine the optimal value of centre points. According to many literatures in setting value of $k$, we predict $k$ following a principle which meets $k \le (\sqrt{n} - 1)/2$ (consider that the optimal value $k$ satisfies the conditions of $k_{\text{opt}} \le k_{\max}$ and $k_{\max} \le \sqrt{n}$). In this clustering algorithm, the solutions equate to the cluster centers. The steps of $K$-means method using ABC algorithm are described as follows.

*Step 1.* Generate the initial population of solutions (the number of solutions is less than the predicted value $k$ mentioned above) and the maximal search times *limit*. Let the $K$-means' cost function (4) be as the objective function.

*Step 2.* Produce new solutions for the employed bees, evaluate them, and apply the greedy selection process.

*Step 3.* Calculate the probabilities of the current sources with which they are preferred by the onlookers.

*Step 4.* Assign onlooker bees to employed bees according to probabilities, produce new solutions, and apply the greedy selection process. That is making a clustering iteration of $K$-means.

*Step 5.* If the search times *Bas* of an employed bee is more than threshold *limit*, stop the exploitation process of the sources abandoned by bees and send the scouts in the search area for discovering new food sources, randomly.

*Step 6.* Memorize the best food source found so far.

*Step 7.* If the termination condition is not satisfied, go to Step 2; otherwise, stop the algorithm.

*Step 8.* Determine the optimal centre points. Then assemble the dataset by these cluster centers and get the final results.

*3.2. Similarity Computation of Modified Collaborative Filtering.* After users clustering, we will use modified collaborative recommendation approach to generate recommendation results for target users. Traditional collaborative filtering method usually adopts the standard cosine similarity or Pearson correlation to compute the similarity between two users. For arbitrary users $u_i$ and $u_j$, the number of common objects shared by them can be expressed as

$$c_{ij} = \sum_{l=1}^{n} a_{li} a_{lj}. \quad (8)$$

Generally, for standard cosine similarity computation, let $s_{ij}$ denote the similarity between $u_i$ and $u_j$ and $k(u_i)/k(u_j)$ denote the degree of the user $u_i/u_j$ namely, how many objects are collected by this user. So we can formulate the expression as:

$$s_{ij} = \frac{c_{ij}}{\sqrt{k(u_i)k(u_j)}} = \frac{\sum_{l=1}^{n} a_{li}a_{lj}}{\sqrt{k(u_i)k(u_j)}}. \tag{9}$$

The problem of (9) is that it has not taken into account the influence of an object's degree, so that objects with different degrees have the same contribution to the similarity. If users $u_i$ and $u_j$ both have selected object $o_l$; that is to say, they have a similar preference for the object $o_l$. In addition, in real recommender system, the similarity computation between two users is not simple but influenced by many factors. So we need to improve the traditional collaborative filtering method in order to fit the complex conditions. In fact, the similarity between two users should be somewhat relative to their degrees and preference degrees; that is, each object node's degree and preference degree are related to its popular degree and corresponding users' comments or ratings, respectively.

We assume that the similarity computation on user-object bipartite networks is affected by an influence degree $1/(1 + |v_{li} - v_{lj}|)k(o_l)$, where $v_{li}/v_{lj}$ represents the preference degree that object $o_l$ obtained from user $u_i/u_j$, $k(o_l)$ denotes the degree of the object $o_l$; namely, how many users select this object. Accordingly, the contribution of object $o_l$ to the similarity $s_{ij}$ should be negatively correlated to its degree $k(o_l)$ and the difference of its preference degrees evaluated by different users. The formulation of $s_{ij}$ can be expressed as

$$s_{ij} = \frac{1}{\sqrt{k(u_i)k(u_j)}} \sum_{l=1}^{n} \frac{a_{li}a_{lj}}{\left(1 + \left|v_{li} - v_{lj}\right|\right)k(o_l)}. \tag{10}$$

Though the users similarity computation, we can find the products unselected by target user but selected by test users who have much similarity to target user. Let $p_{ij}$ represent the preference degree of object $o_j$ obtained from target user $u_i$. The formulation of $p_{ij}$ can be expressed as

$$p_{ij} = \sum_{l=1, l \neq i}^{m} s_{il}a_{jl}. \tag{11}$$

In the process of recommendation, we get the elements of $p_{ij}$ uncollected by target user and then sort them in descending order, as target user prefers the objects in the top, so we recommend top $L$ objects to this user.

*3.3. Recommendation Performance Metrics.* In this paper, we adopt some widely used metrics to measure the accuracy and diversity of the presented recommendation method, in which accuracy is the most important aspect in evaluating the recommendation algorithmic performance. They are five metrics: Ranking score, precision, recall, intrasimilarity, Hamming distance. The first three are used to test accuracy and the rest are used to test diversity. The detailed descriptions of these metrics are as follows.

*(1) Ranking Score.* For an arbitrary user $u_i$, if the recommended object $o_j$ ($o_j$ is an uncollected object for $u_i$) is ranked in $R_{ij}$ position in the ordered recommendation list $L_i$, we can formulate the expression as $r_{ij} = R_{ij}/L_i$. For example, if the length of $L_i$ is 200, namely, there are 200 uncollected objects for $u_i$ and $o_j$ is in the 10th place, we can get the value of $r_{ij} = 10/200 = 0.05$. The average of $r_{ij}$ over all user-object pairs in the test set is denoted by $\langle r \rangle$, which can be used to evaluate the algorithmic accuracy. The smaller the ranking score is, the higher the algorithmic accuracy will be.

*(2) Precision.* It is defined as the ratio of the number of recommended objects collected by users appearing in the test set to the total number of recommended objects. This measure is used to evaluate the validity of a given recommendation list. The precision can be formulated as $a/L$, in which $a$ represents the number of recommended products collected by users appearing in test set, and $L$ is the total number of recommended products. In general, the number of recommended products is no more than 100.

*(3) Recall.* It is defined as the ratio of the number of recommended objects collected by users appearing in the test set to the total number of the objects actually collected by these users. The larger recall corresponds to the better performance. The Recall can be formulated as $a/M$, in which $a$ represents the number of recommended products collected by users appearing in test set, and $M$ is the total number of these users' actual buying.

*(4) Intrasimilarity.* It evaluates the similarity between objects inside users' recommendation lists. A good recommendation algorithm is expected to give fruitful recommendation results and has the ability to guide or help the users exploit their potential interest fields. Therefore, it calls for a lower intrasimilarity. There are many similarity metrics between objects. Here we adopt the widely used one, that is, cosine similarity to measure objects' similarity. For two objects $o_i$ and $o_j$, their similarity is defined as

$$S_{ij} = \frac{\sum_{l=1}^{m} a_{li}a_{lj}}{\sqrt{k(o_i)k(o_j)}}. \tag{12}$$

For an arbitrary user $u_l$, the number of recommendation objects is $L$. Firstly, we need to calculate $L(L-1)/2$ couple of objects' similarity and then average these values to get $I_l = \langle S_{ij} \rangle$. Finally, we use the mean value of $I$ of the overall users to measure the diversity in recommendation lists.

*(5) Hamming Distance.* It can measure the strength of personalization. If the overlapped number of objects in $u_i$ and $u_i$'s recommendation lists is Q, their Hamming distance is

$$H_{ij} = 1 - \frac{Q}{L}. \tag{13}$$

Table 1: The ratio of $K$-means using ABC to traditional $K$-means in metric $D/L$.

| | Number of instances | Number of attributes | $k$ | $D/L$<br>Ratio of $K$-means using ABC to traditional $K$-means |
|---|---|---|---|---|
| Balance | 625 | 4 | 3 | 0.873 |
| Cancer | 569 | 30 | 2 | 0.862 |
| Cancer-Int | 699 | 9 | 2 | 0.871 |
| Credit | 690 | 15 | 2 | 0.891 |
| Dermatology | 366 | 34 | 6 | 0.886 |
| Diabetes | 768 | 8 | 2 | 0.879 |
| Ecoli | 327 | 7 | 5 | 0.881 |
| Glass | 214 | 9 | 6 | 0.895 |
| Heart | 303 | 75 | 2 | 0.868 |
| Horse | 364 | 27 | 3 | 0.859 |
| Iris | 150 | 4 | 3 | 0.860 |
| Thyroid | 215 | 5 | 3 | 0.867 |
| Wine | 178 | 13 | 3 | 0.876 |

What we can know from this table is that our method is better than traditional $K$-means and it has more applicable cluster results.

Generally speaking, a more personalized recommendation list should have long Hamming distances to other lists. Accordingly, we use the mean value of Hamming distance $S = \langle H_{ij} \rangle$ of the overall user-user pairs to measure the strength of personalization.

## 4. Experimental Results and Analysis

In this section, we design two groups of experiments for testing the performance of our method. (1) The first group: we verify the validity of the proposed $K$-means method using ABC algorithm. (2) The second group: we measure the performance of our recommendation approach based on the improved $K$-means. All algorithms mentioned below are implemented in Matlab 7.9 using computer with Intel Core 2 Duo CPU E7500, 2.93 GHz, 4 GB Memory. The operating system of the computer is Windows XP.

*4.1. Validity of K-Means Clustering Method Using ABC.* In fact, Karaboga and Ozturk [27] have demonstrated that ABC algorithm could efficiently be used for multivariate data clustering such as $K$-means. In their work, 13 classification problems from the UCI database, which was a well-known database repository, were used to evaluate the performance of the ABC algorithm. The results showed that $K$-means method using ABC algorithm outperforms it using PSO algorithm in 12 problems, whereas PSO algorithm's result was better than that of ABC algorithm only for one problem (the glass problem) in terms of classification error. Moreover, the average classification error percentages for all problems were 13.13% for ABC and 15.99% for PSO.

Given this, we compare the proposed method with traditional $K$-means in an additional metric $D/L$, where $D$ is the internal distance in clusters and $L$ is the distance between clusters. The smaller the value of $D/L$ is, the better the clustering quality will be. Table 1 shows the results of the comparison.

*4.2. Performance of Recommendation Approach Based on the Improved K-Means.* In testing recommendation algorithmic performance, we use a benchmark dataset *MovieLens* [28] and a real-world dataset. The *MovieLens* dataset consists of 1682 movies, 943 users, and 100,000 ratings. Each user has rated at least 20 movies by using a discrete number on the scale of 1 to 5. In this dataset, there are three kinds of information tables: demographic information about the users, information about the items (movies), and scores about the movies. The real-world data is extracted from a well-known Chinese online bookstore. It contains 86,500 users, providing 1,360,780 ratings about 250,400 books. Each user has an account which records some information from him. The information mainly contains region, age, income, and so on. The ratings are expressed on a discrete number on the scale of 1 to 10. Before experimenting, we need to preprocess these two datasets. For *MovieLens*, only the links with ratings no less than 3 are considered and $v_{li} = \{3, 4, 5\}$. For the real-world dataset, only the links with ratings no less than 5 are considered and $v_{li} = \{5, 6, 7, 8, 9, 10\}$. Besides, we divide the each processed dataset into two parts: the training set which contains 90% of the data and the remaining 10% of the data for the test.

Firstly, the clustering method requires the number of clus ters to be provided as an input. So, we set $k = 15$ for the *MovieLens* and $k = 146$ for the real-world dataset (on the basis of $k = (\sqrt{n} - 1)/2$). Hereinafter, we name our method as cluster-based CF. As known to all, in the steps of generating recommendations, we need to set a certain length of recommendation list $L$ considering recall and precision measures. Generally, merchants recommend no less than 50 ($L$) products to target users through recommender systems.

Table 2: Algorithmic performance for *MovieLens* dataset. The ranking score, precision, recall, intrasimilarity, Hamming distance are corresponding to $L = 30$, 40, and 50. Each number presented in this table is obtained by averaging over five runs, each of which has an independently random division of training set and test.

| Algorithms | Ranking score | Precision | Recall | Intrasimilarity | Hamming distance |
|---|---|---|---|---|---|
| | | | $L = 30$ | | |
| CF | 0.148 | 0.077 | 0.321 | 0.330 | 0.704 |
| MCF | 0.131 | 0.087 | 0.360 | 0.306 | 0.751 |
| NN-CosNgbr | 0.124 | 0.085 | 0.352 | 0.311 | 0.744 |
| cluster-based CF | 0.109 | 0.098 | 0.393 | 0.279 | 0.796 |
| | | | $L = 40$ | | |
| CF | 0.137 | 0.071 | 0.332 | 0.338 | 0.698 |
| MCF | 0.121 | 0.080 | 0.373 | 0.317 | 0.743 |
| NN-CosNgbr | 0.120 | 0.079 | 0.369 | 0.320 | 0.736 |
| cluster-based CF | 0.105 | 0.089 | 0.405 | 0.288 | 0.790 |
| | | | $L = 50$ | | |
| CF | 0.125 | 0.066 | 0.343 | 0.347 | 0.692 |
| MCF | 0.110 | 0.072 | 0.385 | 0.330 | 0.735 |
| NN-CosNgbr | 0.109 | 0.070 | 0.381 | 0.334 | 0.729 |
| cluster-based CF | 0.097 | 0.078 | 0.417 | 0.301 | 0.776 |

Table 3: Algorithmic performance for the real-world dataset. The ranking score, precision, recall, intrasimilarity, Hamming distance are corresponding to $L = 50$, 60, and 70. Each number presented in this table is obtained by averaging over five runs, each of which has an independently random division of training set and test.

| Algorithms | Ranking score | Precision | Recall | Intrasimilarity | Hamming distance |
|---|---|---|---|---|---|
| | | | $L = 50$ | | |
| CF | 0.052 | 0.035 | 0.171 | 0.364 | 0.479 |
| MCF | 0.043 | 0.042 | 0.182 | 0.328 | 0.527 |
| NN-CosNgbr | 0.042 | 0.041 | 0.184 | 0.332 | 0.523 |
| cluster-based CF | 0.034 | 0.047 | 0.201 | 0.272 | 0.614 |
| | | | $L = 60$ | | |
| CF | 0.046 | 0.032 | 0.185 | 0.385 | 0.461 |
| MCF | 0.040 | 0.039 | 0.194 | 0.342 | 0.506 |
| NN-CosNgbr | 0.039 | 0.038 | 0.196 | 0.348 | 0.504 |
| cluster-based CF | 0.031 | 0.043 | 0.209 | 0.281 | 0.603 |
| | | | $L = 70$ | | |
| CF | 0.041 | 0.030 | 0.202 | 0.403 | 0.457 |
| MCF | 0.035 | 0.036 | 0.216 | 0.362 | 0.502 |
| NN-CosNgbr | 0.034 | 0.035 | 0.217 | 0.364 | 0.499 |
| cluster-based CF | 0.027 | 0.040 | 0.231 | 0.293 | 0.592 |

We compare cluster-based CF with three other widely used recommendation algorithms: CF, MCF, and NNCos-Ngbr [29] in all five metrics. We summarize the algorithmic performance in Table 2 for the *MovieLens* and Table 3 for the real-world dataset.

Comparing cluster-based CF with the standard CF, as is seen in Table 2 in the condition of recommendation number $L = 50$, the ranking score can be further reduced by 22.4%, the precision can be further increased by 11.8% and with MCF, the ranking score can be reduced by 18.2%, and the precision can be further increased by 8.3%. Similarly, the proposed algorithm has lower ranking score and higher precision than NN-CosNgbr algorithm. For the rest of metrics, our algorithm is also the best. Although the real-world dataset is similar to *MovieLens*, it is much sparse. So we set the number of recommended products no less than 50.

Table 3 shows that the proposed algorithm exceeds other three algorithms in all the five criterions: lower ranking score, higher precision, bigger recall, lower intrasimilarity, and larger hamming distance.

At last, for an online recommender system, we need to consider the processing time and memory consumption of its recommendation algorithm. In contrast, the process of users clustering increases computational complexity of cluster-based CF. But in the process of similarity calculation, the computing range is significantly reduced that leads to lower computational complexity. Therefore, the computational complexity of cluster-based CF is almost close to traditional CF's. Likewise, the memory store of cluster-based CF is similar to traditional CF's. In addition, the online recommending platform is therefore needed to have efficient access to, at least, two types of resources: data and computing processors. For small scale data recommending tasks, a single desktop computer, which contains hard disk and CPU processors, is sufficient to fulfill the recommending goals. For medium scale data recommending tasks, data are typically large (and possibly distributed) and cannot be fit into the main memory. Furthermore, if a new object is added to the collection or a new user is registered to the recommender system, our algorithm can properly generate recommendation results for it through the clustering computing before recommending.

## 5. Conclusions

Recommendation model helps users to find out their potential future likes and interests. It recommends good products to users and satisfies the users' demands as far as possible. The application of clustering techniques reduces the sparsity and improves the scalability of the recommendation model since the similarity can be calculated only for users in the same clusters. An excellent recommendation method meets high accuracy and certain diversity and can enhance the quality of personalized service.

In this paper, we propose a novel collaborative filtering recommendation approach based on $K$-means clustering algorithm. Firstly, we use artificial bee colony (ABC) algorithm to overcome $K$-means algorithm's problems. And then we adopt the modified cosine similarity considering products' popularity degrees and users' preference degrees to compute the similarity between users in the same clusters. Finally, we generate the recommendation results for target users. Detailed numerical analysis on a benchmark dataset *MovieLens* and a real-world dataset indicates that our new collaborative filtering approach based on users clustering algorithm outperforms many other recommendation methods.

Concerning future work, we will research in the following aspects: how to improve the $K$-means clustering algorithm or adopt other superior clustering algorithms to increase the validity and precision of cluster results. Introduce techniques of implicit information extraction of users and design reasonable clustering partition strategy which considers more actual influence factors; How to keep the robustness of recommendation algorithm when it meets hostile attacks. Hostile attacks mean that someone makes hostile and large number of invalid ratings or evaluations to recommender systems. Through hostile attacks, it is possible to affect the availability of the recommender systems.

## Acknowledgments

This research was supported by National Natural Science Foundation of China (Grant nos. 71071140 and 71071141), Natural Science Foundation of Zhejiang Province (Grant no. LQ12G01007), Ministry of Education, Humanities and Social Sciences Project (Grant no. 13YJCZH216), Foundation of Department of Education of Zhejiang Province (Y201329545), Key Laboratory of Electronic Commerce and Logistics Information Technology of Zhejiang Province (2011E10005), and Innovative Group of e-Business Technology of Zhejiang Province (2010R50041).

## References

[1] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

[2] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 116–142, 2004.

[3] X. Chonghuan, "Personal recommendation using a novel collaborative filtering algorithm in customer relationship management," *Discrete Dynamics in Nature and Society*, vol. 2013, Article ID 739460, 9 pages, 2013.

[4] M. Balabanović and Y. Shoham, "Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[5] M. J. Pazzani, "Framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5, pp. 393–408, 1999.

[6] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[7] J.-Y. Jiang, S.-C. Tsai, and S.-J. Lee, "FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2813–2821, 2012.

[8] G. Adomavicius and Z. Jingjing, "Stability of recommendation algorithms," *ACM Transactions on Internet Technology*, vol. 10, 2012.

[9] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, and F. Yang, "Recommendation model based on opinion diffusion," *Europhysics Letters*, vol. 80, no. 6, Article ID 68003, 2007.

[10] R.-R. Liu, J.-G. Liu, C.-X. Jia, and B.-H. Wang, "Personal recommendation via unequal resource allocation on bipartite networks," *Physica A*, vol. 389, no. 16, pp. 3282–3289, 2010.

[11] S. Maslov and Y.-C. Zhang, "Extracting hidden information from knowledge networks," *Physical Review Letters*, vol. 87, no. 24, Article ID 248701, 2001.

[12] J. Ren, T. Zhou, and Y.-C. Zhang, "Information filtering via self-consistent refinement," *Europhysics Letters*, vol. 82, no. 5, Article ID 58007, 2008.

[13] R.-R. Liu, C.-X. Jia, T. Zhou, D. Sun, and B.-H. Wang, "Personal recommendation via modified collaborative filtering," *Physica A*, vol. 388, no. 4, pp. 462–468, 2009.

[14] H.-N. Kim, I. Ha, K.-S. Lee, G.-S. Jo, and A. El-Saddik, "Collaborative user modeling for enhanced content filtering in

recommender systems," *Decision Support Systems*, vol. 51, no. 4, pp. 772–781, 2011.

[15] M. López-Nores, Y. Blanco-Fernández, J. J. Pazos-Arias, and A. Gil-Solla, "Property-based collaborative filtering for health-aware recommender systems," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7451–7457, 2012.

[16] C.-F. Tsai and C. Hung, "Cluster ensembles in collaborative filtering recommendation," *Applied Soft Computing Journal*, vol. 12, no. 4, pp. 1417–1425, 2012.

[17] I. S. Altingovde, N. Ö. Subakan, and Ö. Ulusoy, "Cluster searching strategies for collaborative recommendation systems," *Information Processing & Management*, vol. 49, no. 3, pp. 688–697, 2013.

[18] H. Wu, X. Wang, Z. Peng, and Q. Li, "Div-clustering: exploring active users for social collaborative recommendation," *Journal of Network and Computer Applications*, vol. 36, no. 6, pp. 1642–1650, 2013.

[19] K. Choi and Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering," *Knowledge-Based Systems*, vol. 37, pp. 146–153, 2013.

[20] M. Gan and R. Jiang, "Constructing a user similarity network to remove adverse influence of popular objects for personalized recommendation," *Expert Systems with Applications*, vol. 40, no. 10, pp. 4044–4053, 2013.

[21] A. Bilge and H. Polat, "A scalable privacy-preserving recommendation scheme via bisecting k-means clustering," *Information Processing & Management*, vol. 49, no. 4, pp. 912–927, 2013.

[22] J. Beringer and E. Hüllermeier, "Online clustering of parallel data streams," *Data and Knowledge Engineering*, vol. 58, no. 2, pp. 180–204, 2006.

[23] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.

[24] B. Basturk and D. Karaboga, "An Artificial Bee Colony (ABC) algorithm for numeric function optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, 2006.

[25] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems," *Lecture Notes in Computer Science*, vol. 4529, pp. 789–798, 2007.

[26] 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.

[27] D. Karaboga and C. Ozturk, "A novel clustering approach: artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 652–657, 2011.

[28] *MovieLens* dataset, http://www.grouplens.org.

[29] M. López-Nores, Y. Blanco-Fernández, J. J. Pazos-Arias, and A. Gil-Solla, "Property-based collaborative filtering for health-aware recommender systems," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7451–7457, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 492615, 16 pages
http://dx.doi.org/10.1155/2013/492615

*Research Article*

# A Location Selection Policy of Live Virtual Machine Migration for Power Saving and Load Balancing

**Jia Zhao,[1,2] Yan Ding,[1,2] Gaochao Xu,[1,2] Liang Hu,[1,2] Yushuang Dong,[1,2] and Xiaodong Fu[1,2]**

[1] *College of Computer Science and Technology, Jilin University, Changchun, Jilin 130000, China*
[2] *Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, Jilin 130000, China*

Correspondence should be addressed to Gaochao Xu; xugc@jlu.edu.cn

Received 26 July 2013; Accepted 13 September 2013

Academic Editors: T. Chen, Q. Cheng, and J. Yang

Green cloud data center has become a research hotspot of virtualized cloud computing architecture. And load balancing has also been one of the most important goals in cloud data centers. Since live virtual machine (VM) migration technology is widely used and studied in cloud computing, we have focused on location selection (migration policy) of live VM migration for power saving and load balancing. We propose a novel approach MOGA-LS, which is a heuristic and self-adaptive multiobjective optimization algorithm based on the improved genetic algorithm (GA). This paper has presented the specific design and implementation of MOGA-LS such as the design of the genetic operators, fitness values, and elitism. We have introduced the Pareto dominance theory and the simulated annealing (SA) idea into MOGA-LS and have presented the specific process to get the final solution, and thus, the whole approach achieves a long-term efficient optimization for power saving and load balancing. The experimental results demonstrate that MOGA-LS evidently reduces the total incremental power consumption and better protects the performance of VM migration and achieves the balancing of system load compared with the existing research. It makes the result of live VM migration more high-effective and meaningful.

## 1. Introduction

VM technology [1, 2], one of the most important technologies in cloud computing, is not only a way to implementing cloud computing such as infrastructure as a service (IaaS) [3] architecture but also the embody of the cloud computing idea, whereas live VM migration technology, which is widely used for the maintenance management in virtualized cloud computing data centers, is the representative of the VM technologies. When a VM needs migrating from source host to target host for some goal or several goals, generally, the migration target of a VM is chosen randomly as long as the host can accommodate it, and then one can automatically or manually move the VM to a target host. It is obvious that the way to randomly choose a target host for a live VM migration, which some event has aroused and has more than one available target host to meet the requirements of that event, is not efficient in all respects. Therefore, a high-efficient location selection policy to migrate the migrant VMs onto the right fit hosts is necessary.

Nowadays, power consumption of data centers has a huge impact on environments. Researchers have been seeking to find effective solutions to minimize power consumption of data centers while keeping the desired quality of service. On the background of low-carbon world and cloud computing era, researchers have already proposed the field of green cloud computing based on cloud computing and virtualization as well as aiming at reducing power consumption in cloud computing data centers.

There are a large number of VMs and tasks running on the hosts of cloud data centers. Some hosts have a heavy load which has a huge impact on the service performance. And some hosts have a relatively lighter load which results in a low utilization of resources. Therefore, it is important to achieve load balancing in cloud data centers as it has covered many key respects of cloud computing data centers.

In this paper, we have focused on the live VM migration policy for power saving and load balancing. Generally speaking, the migrant VM has many available target hosts. However, only one target host is most suitable for the VM in order to achieve the management goals, which include minimizing the total incremental power consumption in cloud data center and achieving load balancing as much as possible in the case that the performance constraint of live VM migration is met.

Many papers have presented some heuristic algorithms to find optimal solutions aiming to minimize power consumption. However, most research has only taken power consumption into consideration but not considered load balancing. In other word, the current research has focused on a single objective optimization for minimizing the power consumption. In this paper, we have focused on the power consumption and load balancing. It is a multiobjective optimization problem (MOP) which differs from the traditional single objective optimization problem. The basic idea of the existing research is that, according to the current situation and history of a cloud data center, the controller has searched for a best migration policy by optimizing the objective function of the proposed algorithm. In the MOPs, the situation is different and complex. It needs to optimize multiple objectives at the same moment. Generally, these objectives influence each other and are even contradictory. To achieve a multiobjective optimization, the problem of how to evaluate and compare two solutions for the multiple functions as well as the problem of how to give each solution a reasonable fitness value towards multiple functions must be addressed first.

It is well known that it is the Pareto dominance principle that is used for MOPs to address the problems. According to the Pareto theory, we know that the optimal solution of an MOP is not a single solution but a set of Pareto optimal solutions. The common and classical heuristic algorithms are used to convert the MOP to a single-objective optimization problem by emphasizing one particular Pareto optimal solution at a time. When such an algorithm will be used to search out multiple solutions, it has to be executed many times, hopefully finding a different solution at each simulation run. However, the genetic algorithm (GA), as the representative of evolution algorithms (EAs), is suitable for dealing with the MOPs since it has the ability to find multiple Pareto optimal solutions in one single simulation run. As EAs work with a population of solutions, a simple EA can be extended to maintain a diverse set of solutions. With an emphasis for moving towards the true Pareto optimal region, an EA can be used to find multiple Pareto optimal solutions in one single simulation run [4, 5]. In this paper, we have proposed a novel heuristic GA-based location selection policy MOGA-LS of VM for power saving and load balancing, which utilizes the Pareto nondominated sorting, individual density estimation, and mathematical statistics of Pareto optimal solutions to achieve a long-term excellent power saving and load balancing optimization. The final experimental results have demonstrated that the MOGA-LS approach not only reduces the total incremental power consumption but also achieves a better load balancing while relatively maximizing the performance of live VM migration. That is, it has reduced the failure events of live VM migration and contributed to achieving better green cloud data centers with load balancing.

The rest of the paper is organized as follows. In Section 2, we present the related work, and the reasonable prerequisites are shown clearly. In Section 3, the main idea, design, problem formulation, solution representation, and implementation of MOGA-LS are introduced in details. In Section 4, the experimental results and analysis on CloudSim platform are given. Finally, in Section 5, we summarize the full paper, and future work is put forward.

## 2. Related Work

As far as we know, the proposed problem which refers to finding a fit target host for a live VM migration according to the standard of minimizing the increment power consumption or load balancing has not been widely researched in the related fields of live VM migration policy let alone both. However, most researchers have focused on some problems which are similar to the proposed problem in this paper. Thus, the related work of the kind of problems relating to live VM migration towards power saving and load balancing will be discussed briefly in this section.

Rusu et al. in [6] have presented a cluster-wide QoS-aware technique that dynamically reconfigures the cluster to reduce energy consumption during periods of reduced load. The proposed system consists of two important components, namely, front end manager and a local manager. While the front end manager finds the servers which should be turned on or off in terms of a given system load, the local manager will utilize dynamic voltage and frequency scaling (DVFS) technique to conserve energy. The main shortage of the approach is the on/off policy. It relies on the table of values and needs offline computing. However, the system does not make use of server consolidation through VM migration, and thus, its on/off policy may not be much effective.

Srikantaiah et al. [7] have investigated the problem of dynamic consolidation of applications serving small stateless requests in data centers to minimize the energy consumption. They modeled the problem as a multidimensional bin packing problem. However, the proposed model doesn't describe the degradation of performance due to the consolidation. Besides, the energy drawn may rely on a particular set of applications combined on a computer node. A heuristic for the defined bin packing problem is proposed by the authors. The heuristic is based upon the idea of minimizing the sum of the current allocations' Euclidean distances to the optimal point at each server. The application workload will be allocated to a server using the proposed heuristic since a request to execute a new application is received. Without the sufficient capacity of all active servers, the system will switch on a new server while reallocating all the applications using the same heuristic in an arbitrary order. The proposed approach is fit for heterogeneous environments; however, it has several shortcomings. First, the approach assumes that all applications' resource requirements are known in advance and constant. Second, performance and energy overhead, which the authors do not take account into, is caused by migration of state-full applications between nodes. The frequent switching servers on/off also generates significant costs which are not negligible for a real-world system.

Verma et al. [8] have contributed energy and migration cost-aware application placement by exploiting the energy management capabilities of virtualization. The authors have designed a new application (virtual machines) placement architecture called pMapper. It consists of three major parts, namely, a performance manager to dynamically resize the VM, an energy manager for CPU throttling, and a migration manager to identify the target host for migration using a knowledge base. They have expounded that for energy-aware scheduling approaches, estimates of energy values are not required, and only if the scheduling algorithm has abilities in finding out which server minimizes the incremental increase in total energy owing to the new VM being placed, it can place the given VM to an appropriate host. In pMapper, two algorithms are implemented. One is First Fit Decreasing (FFD) by which more energy-efficient servers are utilized first without balancing the load. The other is incremental First Fit Decreasing (iFFD) which considers the fixed target utilization of each server and achieves server consolidation by live VM migration. The proposed pMapper architecture minimizes energy and migration costs while ensuring the performance. Our approach is based on a heuristic approach which exploits the concept of minimizing total increase in the incremental energy due to the new VM migrations. The proposed architecture is simple and does not need any knowledge base to achieve significant reduction in the energy consumption.

Li et al. in [9] have proposed an approach named EnaCloud, which enables application of live placement dynamically with consideration of energy efficiency in a cloud platform. In EnaCloud, they use a virtual machine to encapsulate the application, which supports applications' scheduling and live migration to minimize the number of running machines, so as to save energy. In particular, the application placement is abstracted as a bin packing problem, and an energy-aware heuristic algorithm is proposed to get an appropriate solution. In addition, an overprovision approach is presented to deal with the varying resource demands of applications. However, the overprovision approach has risk in optimizing this problem. It may cause more cost in order to reduce cost with a certain probability.

Jeyarani et al. [10] have proposed self-adaptive particle swarm optimization (SAPSO) for efficient virtual machine provisioning in cloud aimed at that when mapping a set of VM instances onto a set of servers from a dynamic resource pool, the total incremental energy drawn upon the mapping is minimal and does not compromise the performance objectives. The advantage of the proposed solution is obvious. It has focused on not only improving the performance of workload facilitating the cloud consumers but also developing the energy efficient data center management to facilitate cloud providers. However, the approach still may be inefficient and may cause some additional events and costs from a long-term perspective as it does not take the future workload into account. Our proposed algorithm MOGA-LS is a heuristic approach which is based on PSO, one of swarm intelligence algorithms and introduces the simulated annealing (SA) idea into it.

Jing and She [11] have proposed a novel model for this problem of live VM migration for load balancing based on fuzzy TOPSIS to detect the hotspots and balance load. The proposed model is to migrate VMs between hosts using fuzzy TOPSIS theory to make decision over the whole of active hosts of the data center and detect the hotspots. The proposed model can be a suitable tool to rank hosts. However, the kind of load balancing policies based on sorting hosts is not heuristic enough and has a relatively complex computation. It is not efficient for achieving load balancing.

Yang et al. in [12] have proposed a load balancing algorithm based on live VM migration. The proposed algorithm includes two major policies: trigger strategy based on the load prediction and selection strategy of the destination node based on the multiple criteria decision. Yet the kind of load balancing policies based on load predicting is not accurate enough and needs to maintain additional historical data, as results in the unnecessary system load.

In this paper, MOGA-LS has a prerequisite. We know that MOGA-LS is to find the target host of each VM from all $m$ hosts for the $n$ migrant VMs. We assume that each VM's target host found by MOGA-LS will not be the host which the VM is moved out from. The algorithm provides service to live VM migration aiming at green cloud data center with load balancing. Thus, the fact that the VM should be moved out for some reason is the premise of our approach. Objectively, the prerequisite is justified from a certain perspective. Since a VM needs to be migrated from its source host, its candidate hosts will not include its source host. Otherwise, it does not need a migration event. It can be seen that for all the migrant VMs, the hosts each of which is the source host of some migrant VM will not be the target hosts. Therefore, the proposed prerequisite is reasonable and does not impact the performance and efficiency of our approach.

## 3. Proposed MOGA-LS Approach

*3.1. Main Idea.* The proposed MOGA-LS approach is a heuristic live VM migration policy, which is used for the target location selection of live VM migration. It is a multiobjective optimization algorithm which is based on the improved GA that utilizes Pareto theory to achieve the selection, crossover, and mutation operators of GA towards multiple objectives and thus to find out the set of Pareto optimal solutions of the proposed multiobjective problem by the evolution of the population. After obtaining the set of Pareto optimal solutions, in order to get the final migration policy, we do not randomly select a solution but utilize the mathematical statistics theory to obtain the final solution of location selection of live VM migration by using a probability wheel for each VM.

Specifically, we have employed a novel policy which is used to generate the initial population and defined a Pareto constraint dominance relation towards the proposed constrained problem to compare two solutions in the improved

GA-based approach. The nondominated sorting policy and density estimation policy used for the population has been presented to make each individual of the population obtain fitness values in each generation. We have employed the tournament selection operator for the selection operator of GA. For the crossover and mutation operators, we have designed the arithmetic crossover operator and the dynamic nonuniform mutation operator. Besides, the $(\mu + \lambda)$ selection policy, which makes elitism that helps in achieving better convergence be introduced into the proposed MOGA-LS approach, is employed to generate the next population.

*3.2. Design of the MOGA-LS Approach.* The MOGA-LS approach is an algorithm based on multiobjective GA achieving a live VM migration policy for minimizing the incremental power consumption caused by migrating these migrant VMs onto their target hosts and making the load of cloud data center balanced after migrating under the constraint of maximizing the performance that the number of success of live VM migration events is maximized. To achieve a multiobjective GA, we have introduced the concept of Pareto optimal solutions into GA and designed a constrained Pareto dominance method to evaluate the individuals of population and assign fitness values to them as well as designing GA's genetic operators including selection, crossover, and mutation operators as well as the policy of generating the new population.

*3.2.1. Problem Formulation.* We now formulate the proposed problem of migrating $n$ VMs onto $m$ hosts. Its solution can be represented by an $n$ dimension of vector, each element of which denotes the target host of the migrant VM which its location represents. We assume that there are $m$ available hosts in the resource pool of a cloud data center and the hosts are heterogeneous and dynamic while using space shared allocation policy. The hosts change their state dynamically according to the load. Similarly, each VM is associated with the required computing resource. It is considered that the VMs are independent of each other and are equal in priority.

The problem can be stated as follows. Find a VM-host mapping set LS of location selections such that the total incremental power consumption caused by the migrated VMs onto hosts is minimized and after migrating the load is balanced as much as possible, while maximizing the performance by fulfilling the resource requirements of maximum number of VMs. We define a quintuple $S = \{\text{PH}, \text{VM}, \text{PC}, \text{RL}, \text{LS}\}$ for our problem scenario. PH is a set of $m$ available physical hosts denoted by PH $(m, t) = \{\text{PH}1, \text{PH}2, \text{PH}3 \ldots, \text{PH}m\}$, available at migrating start time $t$. VM is a set of $n$ migrant VMs denoted by VM $(n, t, \Delta t) = \{\text{VM}1, \text{VM}2, \text{VM}3, \ldots, \text{VM}n\}$ accumulated within a time window $\Delta t$. PC $(m, t) = \{P1, P2, \ldots, Pm\}$ is the power consumption by the $m$ physical hosts in the resource pool. RL $(m, t) = \{\{R_{\text{CPU}}1, R_{\text{MEM}}1\}, \{R_{\text{CPU}}2, R_{\text{MEM}}2\}, \{R_{\text{CPU}}3, R_{\text{MEM}}3\}, \ldots, \{R_{\text{CPU}}m, R_{\text{MEM}}m\}\}$ represents the residual CPU and memory resource of each host at migrating start time $t$. The problem is multimodal, generally having more than one location selection which meets the performance constraint of the VM requests. Our goal is to find a location selection which meets the performance constraint while minimizing the incremental power consumption and minimizing the standard deviation of the residual load rates of all available hosts to achieve load balancing. That is, the goal is to obtain all Pareto optimal solutions. It can be seen as a constrained multiobjective Pareto optimization problem.

To fulfill the performance constraint, a metric denoted by $\eta_r$ representing resource fulfillment requirement, $\eta_r$ represents the number of success of live VM migration events and should be equal to $n$ as a constraint function ideally, is defined as follows:

$$\eta_r = \sum_{i=1}^{n} \sum_{j=1}^{m} t_i^j = n, \quad i \in \{1, 2, 3, \ldots, n\}, \; j \in \{1, 2, 3, \ldots, m\}, \quad r \in \{1, 2, 3, \ldots, s\}, \tag{1}$$

where $\tau_i^j$ denotes the location selection of $i$th VM on the $j$th host and is defined as follows, $s$ denotes the total number of location selections:

$$\tau_i^j = \begin{cases} 1 \\ \text{if VM}_i \text{ has been allocated to} \\ \text{Host}_j \text{ and rcrVM}_i \leq \text{acrHost}_j, \\ \qquad i \in \{1, 2, 3, \ldots, n\} \\ 0 \\ \text{if VM}_i \text{ has been allocated to} \\ \text{Host}_j \text{ and rcrVM}_i \leq \text{acrHost}_j, \\ \qquad j \in \{1, 2, 3, \ldots, m\} \\ \text{Invalid} \\ \text{if VM}_i \text{ has been allocated to Host}_j. \end{cases} \tag{2}$$

In (2), rcrVM$_i$ denotes the minimum computing resource requirements of $i$th VM and acrHost$_j$ denotes the available computing resource of $j$th host.

The first objective function is based on power consumption. The migration of successive VMs is represented as LS $[m, n, r, t + t_0(k)]$ where $r$ represents any one of $s$ location selections and $k$ is an integer increasing with successive migrating, representing a stage. We can understand that if $k$ is 3, the approach will migrate the third VM to the host which is denoted at the third location of the $r$ vector. The $s$ location selections at stage $k$ are represented as LS $[m, n, r, t{+}t_0(k)]$ $r \in \{1, 2, 3, \ldots, s\}$ and their corresponding power consumption is represented as $\xi_k^r$. Its meaning is that according to $r$ location vector after the system has migrated the $k$th VM to its target host, the total power consumption by the cloud data center is $\xi_k^r$, ideally. The meaning of $\xi_{k-1}^r$ can be imaged. In this paper, these parameters of $\xi_k^r$ can be obtained by using simulation platform in the experiment. Now, the incremental power consumption due to migrating LS $[m, n, r, t + t_0(k)]$ with respect to previous migration stage LS $[m, n, r, t + t_0(k - 1)]$ is defined by

$$\Delta P = (\xi_k^r) - (\xi_{k-1}^r); \quad r \in \{1, 2, 3, \ldots, s\}. \tag{3}$$

For better power saving, the following $\delta P^r$ should be minimized, and it is denoted as follows:

$$\delta P^r = \sum_{k=1}^{n} \left( \left( \xi_k^r \right) - \left( \xi_{k-1}^r \right) \right); \quad r \in \{1, 2, 3, \ldots, s\}. \tag{4}$$

Therefore, the proposed MOGA-LS approach minimizes $\delta P^r$ for power efficiency. Function (4) is the first objective for MOGA-LS.

The second objective function is based on load balancing. In this paper, we have presented the residual load rate to measure the load situation of each host. The calculation method of the residual load rate is described as follows. Now, assume that the set of available hosts is PH $(m, t) = \{\text{PH1}, \text{PH2}, \text{PH3} \ldots, \text{PH}m\}$ after a time window $\Delta t$. Within the $\Delta t$, the set of the accumulated live VM migration requests is represented as VM $(n, t, \Delta t) = \{\text{VM1}, \text{VM2}, \text{VM3}, \ldots, \text{VM}n\}$. After migrating all the migrant VMs based on a location selection $r \in \{1, 2, 3, 4, 5, 6, \ldots, s\}$, the residual load $R^{ir}$ of the host $i$ is defined as follows:

$$\begin{gathered} R^{ir} = \alpha R_{\text{CPU}}^{ir} + \beta R_{\text{MEM}}^{ir}, \quad i \in \{1, 2, 3, \ldots, m\}, \\ r \in \{1, 2, 3, \ldots, s\}, \\ \alpha + \beta = 1, \end{gathered} \tag{5}$$

where $R_{\text{CPU}}^{ir}$ and $R_{\text{MEM}}^{ir}$ represent the residual CPU and memory resource of host $i$ after migrating according to the solution vector $r$. So, the residual load rate $E^{ir}$ of host $i$ is defined as follows:

$$E^{ir} = \frac{R^{ir}}{T^i}, \quad i \in \{1, 2, 3, \ldots, m\}, \; r \in \{1, 2, 3, \ldots, s\}, \tag{6}$$

where $T^i$ can be represented as follows:

$$T^i = \alpha T_{\text{CPU}}^{i} + \beta T_{\text{MEM}}^{i}, \quad i \in \{1, 2, 3, \ldots, m\}, \tag{7}$$

where $T_{\text{CPU}}^{i}$ denotes the total CPU resource of host $i$ and $T_{\text{MEM}}^{i}$ denotes the total memory resource of host $i$. In this paper, we think that, in order to make the load of all $m$ physical hosts balanced as much as possible, the residual load rate $E^{ir}$ of each host should be as similar as possible after having migrating all migrant VMs accumulated within a time window $\Delta t$. Therefore, we have utilized the standard deviation of all hosts' residual load rates to formulate this problem. The formula of the expectation and the standard deviation is as follows:

$$\begin{gathered} E(X) = \frac{\sum_{i=1}^{N} X_i}{N}, \\ \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left( X_i - E(X) \right)^2}. \end{gathered} \tag{8}$$

By using the above two formulas, the second objective function can be described as follows:

$$\sigma^r = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left( \frac{\alpha R_{\text{CPU}}^{ir} + \beta R_{\text{MEM}}^{ir}}{\alpha T_{\text{CPU}}^{i} + \beta T_{\text{MEM}}^{i}} - \frac{\sum_{k=1}^{m} \left( \left( \alpha R_{\text{CPU}}^{kr} + \beta R_{\text{MEM}}^{kr} \right) / \left( \alpha T_{\text{CPU}}^{k} + \beta T_{\text{MEM}}^{k} \right) \right)}{m} \right)^2}. \tag{9}$$

So far, we have formulated the proposed problem as a multiobjective optimization problem with a constraint. That is,

$$\begin{aligned} \text{Min} \quad & \begin{cases} \delta P^r = \sum_{k=1}^{n} \left( \left( \xi_k^r \right) - \left( \xi_{k-1}^r \right) \right), \\ \sigma^r = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left( \frac{\alpha R_{\text{CPU}}^{ir} + \beta R_{\text{MEM}}^{ir}}{\alpha T_{\text{CPU}}^{i} + \beta T_{\text{MEM}}^{i}} - \frac{\sum_{k=1}^{m} \left( \left( \alpha R_{\text{CPU}}^{kr} + \beta R_{\text{MEM}}^{kr} \right) / \left( \alpha T_{\text{CPU}}^{k} + \beta T_{\text{MEM}}^{k} \right) \right)}{m} \right)^2}, \\ \qquad r \in \{1, 2, 3, \ldots, s\} \end{cases} \\ \text{s.t.} \quad & \eta_r = \sum_{i=1}^{n} \sum_{j=1}^{m} \phi_i^j = n, \quad i \in \{1, 2, 3, \ldots, n\}, \; j \in \{1, 2, 3, \ldots, m\}, \; r \in \{1, 2, 3, \ldots, s\}. \end{aligned} \tag{10}$$

*3.2.2. Relevant Concepts of Pareto Optimal Solutions.* In a MOP, the fitness values cannot be compared between multiple objectives. And generally these objectives are in conflict with each other. The improvement of an objective is often at the expense of impairing the fitness value of the other objective. Thus, it can be seen that the solution of a MOP is not only one but a set consisting of many solutions which cannot compare with each other. These solutions are called Pareto solutions set. In all Pareto solutions sets of a MOP, the best one is Pareto optimal solutions set. To explain this problem, we have given some definitions in the following.

*Pareto Dominance.* That a solution vector $u = (u_1, u_2, \ldots, u_n)$ dominates the other solution vector $v = (v_1, v_2, \ldots, v_n)$ ($u \prec v$) refers to that for all $i \in \{1, 2, 3, \ldots, k\}$ $f_i(u) \leqslant f_i(v)$ and $\exists j \in \{1, 2, 3, \ldots, k\}$ s.t. $f_j(u) < f_j(v)$. In the definition, the MOP is a minimization problem by default.

*Pareto Optimal Solutions.* That a candidate solution $X = (x_1, x_2, \ldots, x_k) \in \Omega$ is a Pareto optimal solution refers to that $\neg\exists X' \in \Omega$ s.t. $X' \prec X$.

*Pareto Optimal Solutions Set.* $P = \{X \in \Omega \mid \neg\exists X' \in \Omega : X' \prec X\}$.

*Pareto Front.* $\mathrm{PF} = \{F(X) = (f_1(X), f_2(X), \ldots, f_k(X)) \mid X \in P\}$.

As can be seen from these definitions, our goal is to find Pareto optimal solutions set. That is, the MOGA-LS approach should return the Pareto optimal solutions set finally. We have utilized Pareto dominance to compare any two individuals of population. As our proposed problem has a constraint, we have redefined the Pareto dominance as a constrained Pareto dominance as follows. A solution $i$ is said to constrained-dominate a solution $j$, if any of the following conditions is true.

(1) Solution $i$ is feasible and solution $j$ is not.

(2) Solutions $i$ and $j$ are both infeasible, but solution $i$ has a smaller overall constraint violation.

In the proposed problem, this scenario refers to that solution $j$ results in more failure events of live VM migration than solution $i$. This means that if neither of the two solutions can make all live VM migration events accumulated within a time window $\Delta t$ successful, the solution which results in less failure events of live VM migration is better since that the most migrant VMs are migrated successfully is the premise of the proposed problem. After all, the MOGA-LS approach is a live VM migration policy.

(3) Solutions $i$ and $j$ are feasible and solution $i$ dominates solution $j$.

In this paper, the Pareto dominance mentioned by the proposed MOGA-LS approach refers to the newly defined constrained Pareto dominance.

*3.2.3. Solution Representation and GA Encoding.* In order to design an efficient GA-based algorithm for finding the optimal vector of the target hosts of all migrant VMs in a time window $\Delta t$, the primary problem is the solution representation as it represents a direct relationship between the problem domain and the individuals in GA. During the applying of a GA-based algorithm, we know that an individual is denoted as a solution of the specific problem. Here, there are $n$ VMs to be migrated into $m$ physical hosts. So, the proposed problem is an $n$ dimensional problem, leading to that the individuals are represented as some $n$ dimensional vectors. Each dimension has a discrete set of all possible location selections which are denoted as integers and limited to $m$. A solution vector called an individual or a chromosome is denoted by $x_i^k = (x_{i1}^k, x_{i2}^k, \ldots, x_{ij}^k, \ldots, x_{iD}^k)$, where the $x_{ij}^k$ value represents the coded value (the no. of target host of the $j$th migrant VM) of the $j$th gene (the $j$th migrant VM) of $i$th individual (the $i$th possible solution vector) in $k$th generation.

*3.2.4. Design of Genetic Operators in MOGA-LS*

*(1) Selection Operator.* In the MOGA-LS approach, we have employed the tournament selection operator. Our main idea is that the algorithm randomly chooses two groups of individuals from the population. Each group consists of $k$ individuals. From the efficiency and diversity of the MOGA-LS approach to consider, the tournament scale $k$ is set to 2 in this paper. That is, the algorithm randomly chooses two groups, each of which includes two individuals from the original population. The two winning individuals of the two groups are obtained by the comparison within groups. In the next step, the two individuals will be used for the crossover operator.

*(2) Crossover Operator.* As mentioned above, the GA encoding has employed the positive integer coding, and the integer is limited to $m$. We have not utilized the widely used Simulated Binary Crossover (SBX) crossover method that for a random given crossover point, the two parent individuals exchange the sections located on both sides of the crossover point. However, we have designed the nonuniform arithmetic crossover operator and introduced it into our approach in order to improve the global search ability and better keep the diversity of population. Let $X_i^t$ and $X_j^t$, respectively, represent the real encoding values of the crossover points of the two parent individuals $i$ and $j$ in the $t$th generation. After the crossover, the corresponding gene encoding values $X_i^{t+1}$ and $X_j^{t+1}$ of the two individuals are defined as follows:

$$\begin{aligned} X_i^{t+1} &= \alpha X_i^t + (1-\alpha) X_j^t, \\ X_j^{t+1} &= (1-\alpha) X_i^t + \alpha X_j^t, \end{aligned} \tag{11}$$

where $\alpha$ is a parameter and is not a constant. It is related to the evolution generation number. The specific definition of $\alpha$ in this paper will be described in the following.

*(3) Mutation Operator.* MOGA-LS is a heuristic approach based on GA. As a heuristic optimization algorithm, it should have better global search ability in the early iterations, and it should have better local search and convergence ability in the later iterations. Therefore, we have utilized the dynamic nonuniform mutation operator to make the scope of the gene mutation change with the increase of the generation number and thus to improve the search and convergence ability of MOGA-LS. Now, we assume that an individual (a chromosome) $X = (X_1, X_2, \ldots, X_k, \ldots, X_n)$ is mutated to a new individual $X' = (X_1, X_2, \ldots, X_k', \ldots, X_n)$. If the gene encoding value $X_k$ of the mutation point $k$ ranges within

$[U_{\min}^{k}, U_{\max}^{k}]$, the new gene $X_k'$ is determined by the following formula:

$$X_k' = \begin{cases} X_k + \left(U_{\max}^{k} - X_k\right)\left(1 - \gamma_k^{(1-(t/T))^{\varepsilon}}\right), \\ \qquad \text{if } r \text{ and } (0,1) \in [0, 0.5), \\ X_k - \left(X_k - U_{\min}^{k}\right)\left(1 - \gamma_k^{(1-(t/T))^{\varepsilon}}\right), \\ \qquad \text{if } r \text{ and } (0,1) \in [0.5, 1], \end{cases} \tag{12}$$

where $\gamma_k$ is a random number distributed uniformly between 0 and 1. $T$ is the maximum number of iterations of MOGA-LS. $t$ is the current number of iterations of MOGA-LS. $\varepsilon$ is a system parameter which is used to determine dependency degree on the number of iterations.

It is obvious that $\Delta t = (U_{\max}^{k} - X_k)(1 - \gamma_k^{(1-(t/T))^{\varepsilon}})$ ranges within $[0, (U_{\max}^{k} - X_k))$. To begin with, $t$ is smaller, and thus, $\Delta t$ is larger, making the gene value have an obvious mutation. With the increase of the number $t$ of iterations, the value of $\Delta t$ becomes gradually smaller. The change of the gene value becomes smaller. This feature makes the operator have ability in searching the whole space evenly in the early iterations (when $t$ is small) and precisely searching several partial areas in the later iterations. To say further, the mutation operator makes the MOGA-LS approach have better global search ability in an early phase and have better local search ability and convergence in a later phase as it is a dynamic and self-adaptive mutation operator which can be adjusted by modifying the parameters such as $\gamma_k$ and $\varepsilon$.

### *3.2.5. Design of Fitness Values in MOGA-LS*

*(1) The Fitness Value $r_p$.* The design of the fitness values is the core of GA. And especially for multiobjective GA, it is more important since this kind of problems has multiple objective functions. In this paper, the Pareto dominance is utilized to achieve the Pareto ranking approaches and thus to obtain the fitness values of each individual in each generation.

As mentioned above, the MOPs are the Pareto optimization problems, which have employed the Pareto dominance to compare and evaluate the individuals. The population is ranked according to the Pareto dominance rule, and then each solution is assigned a fitness value based on its rank in the population, not any one of its actual objective function values. Note that herein a lower rank corresponds to a better solution. The rank of each individual refers to its nondominated rank and is called $r_p$. Each individual has a parameter $n_p$, whichis the number of individuals that dominate the individual $p$. Each individual can be compared with every other individual in the population to find if it is dominated.

The rank $r_p$ of each of the individuals whose $n_p$ values are 0 is set as 1. At this stage, all individuals in the first nondominated front are found. In order to find the individuals in the next nondominated front, the solutions of the first front are discounted temporarily and the above procedure is repeated. The rank $r_p$ of each of the individuals whose $n_p$ values are 0 is set as 2, and so forth, until all the individuals of the population are ranked in this generation. After the nondominated sorting, the population is divided into several ranks and each individual has a rank $r_p$. In other words, each of these ranks is a set consisting of several individuals.

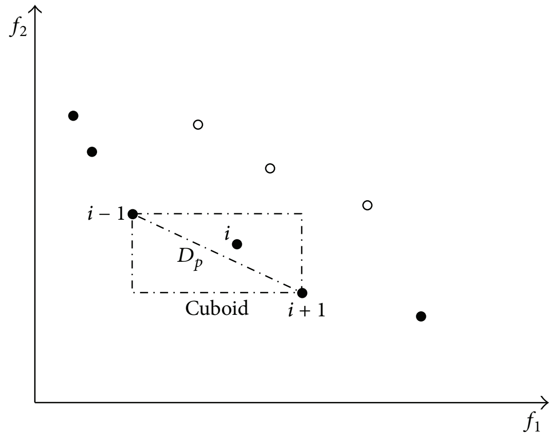


FIGURE 1: Crowding-distance calculation of the density estimation.

*(2) The Fitness Value $D_p$.* The individuals between these ranks are comparable. In the same rank, the individuals cannot be compared with each other and are equal for being selected. Furthermore, we know that maintaining a diverse population is an important consideration in multiobjective GA to obtain solutions uniformly distributed over the Pareto optimal solutions set. Without taking preventive measures, the population tends to form relatively few clusters in multiobjective GA. This phenomenon which will prevent the MOGA-LS approach convergent to Pareto optimal solutions set is called genetic drift, in genetics. To further address the two problems of the fitness values and the diversity of a population, we have designed and employed the density estimation approach aiming to obtain a uniform spread of solutions along the Pareto front. Its main idea is that in the same rank the individuals in which there are fewer individuals around are better than others. It should be selected more potentially to generate the next generation in order to prevent genetic drift and keep the diversity of the population and thus to optimize the MOGA-LS approach.

The process of the density estimation approach is as follows. To get an estimate of the density of solutions surrounding a particular solution in the population, we need to calculate the distance of two points on either side of this point along each of the objectives. This quantity $D_p$ serves as an estimate of the diagonal of the cuboid formed by using the nearest neighbors as the vertices (call this the crowding distance). In this paper, we have defined the crowding distance of the $i$th solution in its front as the diagonal length of the cuboid as shown in Figure 1. The computation of the crowding distance needs to sort the population according to each objective function value in ascending order. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the corresponding diagonal length. If a MOP has three objective functions, the crowed distance is the diagonal length of a cube. If a MOP has two objective functions, the crowed distance is the diagonal length of a rectangle, and so forth. Each objective function is normalized before calculating the crowding distance.

*(3) The Fitness Value $R_p$.* Now, each individual in the population has two attributes $r_p$ and $D_p$ in each generation. The Pareto rank $r_p$ is the core of the MOGA-LS approach. Its convergence process is the process of converging the nondominated set of each generation to the Pareto optimal solutions set of the final generation. A rank represents a set of individuals where each element belongs to the nondominated rank. In other words, the individuals in the same rank are equal and have the same chances to be selected for generating the next generation. However, as a matter of fact, the individuals in the same rank are not equal due to their different situations of being dominated, especially for the individuals located in the ranks larger than 1. For each individual in the same rank, their $r_p$ values are naturally equal. However, the $r_p$ value does not reflect the ranks of the individuals dominating it. It is obvious that if the sum of the rank values of the individuals that dominate an individual is larger than that of other individuals in the same rank, the individual is not better for being used to generate the next generation. This is because its gene is not excellent enough and it is not beneficial for keeping the diversity of the population. Thus, the information included in the $r_p$ value is not enough. We have presented a novel parameter $R_p$. It is defined as follows:

$$R_p = 1 + \sum_{j=1}^{t} \gamma_{X_j}, \tag{13}$$

where we assume that the individual $p$ is dominated by $t$ individuals $X1, X2, \ldots, Xt$, whose rank values are already known as $\gamma_{X1}, \gamma_{X2}, \ldots, \gamma_{Xt}$. The dominated individuals are penalized to reduce the population density and redundancy. Here, we have given an instance. The multiobjective GA generates 11 individuals. Their rank values are illustrated in Figure 2. Suppose we want to minimize two objectives $f_1$ and $f_2$. Both $A$ and $B$ belong to rank 2. However, the $R_p$ value of individual $A$ is 5, and the $R_p$ value of individual $B$ is 2. The individual $B$ is better. What is more, for the individuals that have the same ranks or the different ranks, $R_p$ is more suitable as a fitness value for comparison since it has included not only the information of the nondominated rank of the individual but also the information of it being dominated for penalizing it. Meanwhile, it can be seen that the $R_p$ value has also better kept the diversity of the population. From this perspective, it also makes up for the shortcoming of the density estimation since the density estimation approach is limited to the same rank. In order to achieve the calculation of $R_p$, each individual $i$ has a parameter $S_p$, which is a set consisting of the $r_p$ values of the individuals dominating the individual $i$.

Thus far, in our MOGA-LS approach, each individual of the population has 3 attributes $R_p$, $r_p$, and $D_p$ as its fitness values in each generation. It can be seen that our design is that when comparing two individuals, the algorithm compares first their $R_p$ values, second their $r_p$ values, and last their $D_p$ values. We now define a partial order $\prec_{R_p,\gamma_p,D_p}$ as follows:

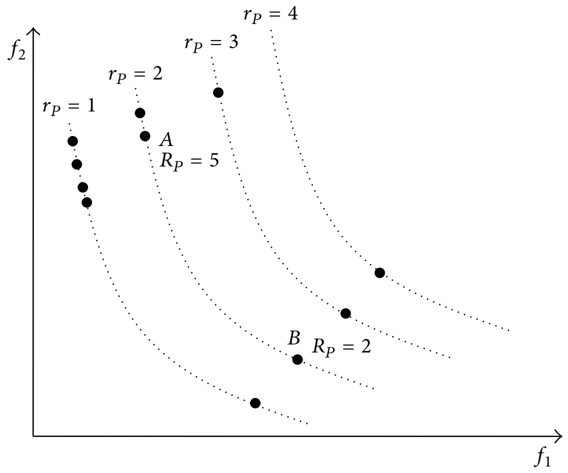


Figure 2: Individual rank values.

$$\begin{aligned} i \prec_{R_p,\gamma_p,D_p} j \quad & \text{if } \left(i_{R_p} < j_{R_p}\right) \\ & \text{or } \left(\left(i_{R_p} = j_{R_p}\right), \left(i_{\gamma_p} < j_{\gamma_p}\right)\right) \\ & \text{or } \left(\left(i_{R_p} = j_{R_p}\right), \left(i_{\gamma_p} = j_{\gamma_p}\right), \right. \\ & \left. \left(i_{D_p} > j_{D_p}\right)\right). \end{aligned} \tag{14}$$

The design not only has utilized the Pareto theory to achieve multiobjective comparison but also kept the diversity of the population, avoided genetic drift, and made the population more efficiently converged to the Pareto optimal solutions set with no additional complexity.

*3.2.6. System Architecture of MOGA-LS.* Figure 3 depicts the proposed architecture for cloud environment. It shows the position of the controller MOGA-LS for migration location selections and its interaction with other entities [13, 14]. In a time window $\Delta t$, the monitor gets the requests of live VM migration and updated with the available number of computing resource such as CPUs, memory, and storage, as well as power consumption. At the end of $\Delta t$, the monitor transfers the information to the controller. The controller generates the location policy by using the proposed approach and obtained information. Then, it transfers the policy to the migration controller which controls and executes live migration of the VMs. The VMs are eventually moved onto their target hosts.

*3.3. Implementation of MOGA-LS.* In this section, we describe the specific process of MOGA-LS. Details are as follows.

*Initialize the MOGA-LS Algorithm.* The size of the population is $N$. The number of the genes of an individual (a chromosome) is $n$ due to the $n$ migrant VMs as mentioned above.

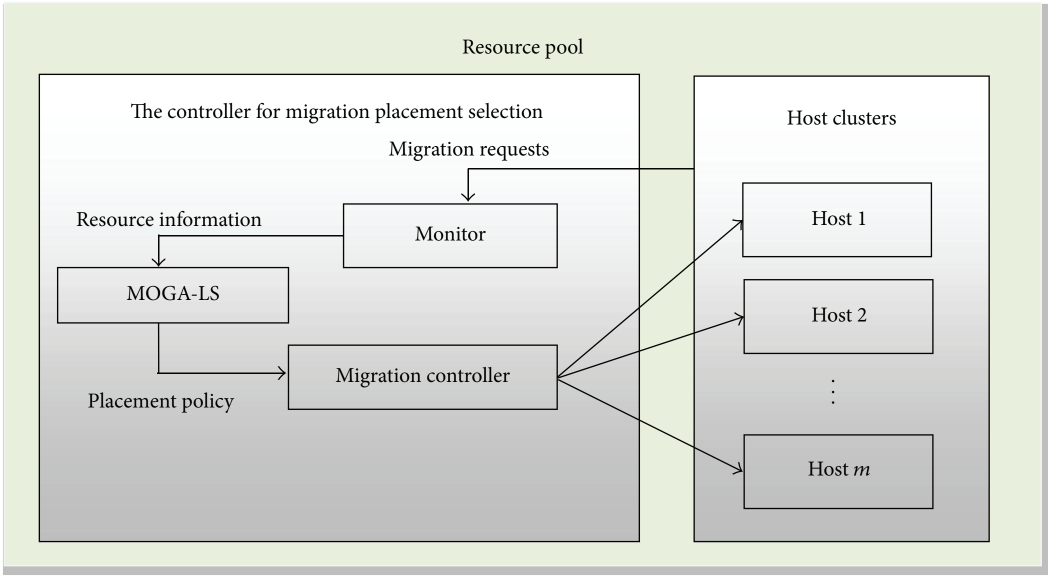


FIGURE 3: The view of MOGA-LS's architecture.

The maximum number of iterations (evolution) is $i_{\max}$. The mutation rate of the population is $M_r$. The competition scale of the tournament selection operator is set to 2. The initial population $P_0$ is generated by employing the way in which the initial *archive* is generated in [15].

*Evolve the Population of MOGA-LS.* Each individual of $P_0$ can be compared with every other solution to find if it is dominated and thus to obtain the $n_p$ value of each individual by using the constrained fitness functions group (10). The algorithm sets the $\gamma_p$ values of the individuals whose $n_p$ values are 0 as 1. At this stage, all individuals in the first nondominated front are found. In order to find the individuals in the next nondominated front, the solutions of the first front are discounted temporarily and the above procedure is repeated. The rank $r_p$ of each of the remaining individuals whose $n_p$ values are 0 is set as 2, and so forth, until all the individuals of the population are ranked in this generation.

After the nondominated sorting, the population is divided into several ranks and each individual has a rank value $r_p$. Thereafter, each individual of $P_0$ obtains its $R_p$ value by using (13) and its $S_p$ set. The algorithm first sorts the population $P_0$ according to the first objective function (4) and the second objective function (9) in ascending order, respectively. The boundary solutions with smallest and largest function values are assigned an infinite $D_p$ value. All other intermediate solutions are assigned a distance value $D_p$ as follows:

$$i_{D_p} = \sqrt{\left(\frac{(i+1)\cdot f_1 - (i-1)\cdot f_1}{f_1^{\max} - f_1^{\min}}\right)^2 + \left(\frac{(i+1)\cdot f_2 - (i-1)\cdot f_2}{f_2^{\max} - f_2^{\min}}\right)^2}. \quad (15)$$

Now each individual of the initial population $P_0$ has the fitness values $R_p$, $\gamma_p$, and $D_p$. The selection operator of MOGA-LS begins to perform the tournament selection of $P_0$. Two groups of individuals are chosen randomly. Every group consists of two individuals. In each group, the individual with better fitness values is picked out according to the proposed partial order $r : \prec_{R_p,\gamma_p,D_p}$. The crossover operator of MOGA-LS begins to perform the arithmetic crossover operation for the two winning individuals of the two groups according to (11) and thus to obtain two new individuals. The information of the Pareto nondominated rank values of the individuals of the population is introduced into the arithmetic crossover operator used by us. In the arithmetic crossover formula (11), the parameter $\alpha$ is defined as follows:

$$\alpha = \frac{j_{\gamma_p}}{i_{\gamma_p} + j_{\gamma_p}}. \quad (16)$$

The crossover operator coefficient $\alpha$ is associated with the nondominated rank of each individual in the population. In the early iterations of MOGA-LS, the $\alpha$ value will have a great change. However, with the evolution of the population, the individuals of the population will tend to the same Pareto front and the $\alpha$ value will tend to the constant 0.5.

Repeat the selection and crossover operations until generating a population of size $N$. The mutation operator of MOGA-LS performs the dynamic nonuniform mutation for the generated population according to the mutation rate and (12) to eventually obtain the offspring $Q_0$ of $P_0$. At this time, the MOGA-LS algorithm forms a combined population $R_0 = P_0 \cup Q_0$. The size of the population $R_0$ is $2N$. Then, the population $R_0$ is sorted according to nondomination rank. The elitism is ensured as all previous and current population members are included in $R_0$.

Now, the individuals belonging to the best nondominated set $S_1$ are the best solutions in the combined population and

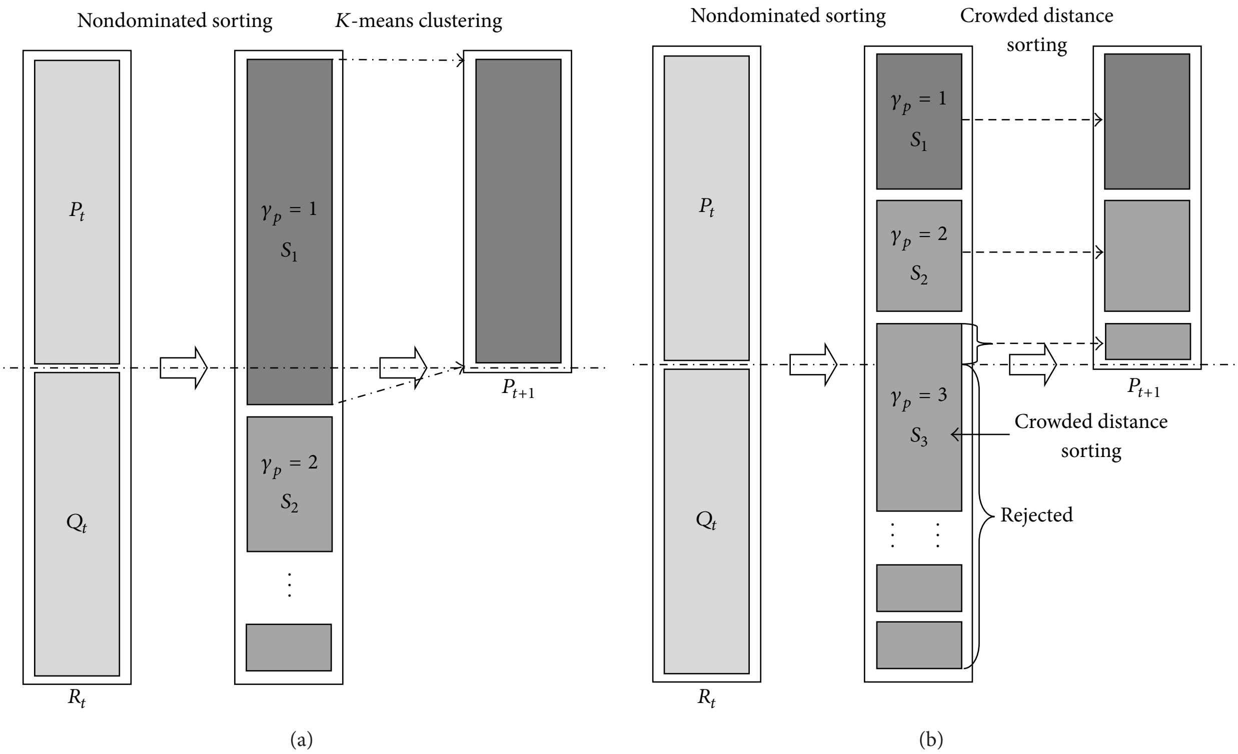


Figure 4: The procedure of generating the new population in MOGA-LS.

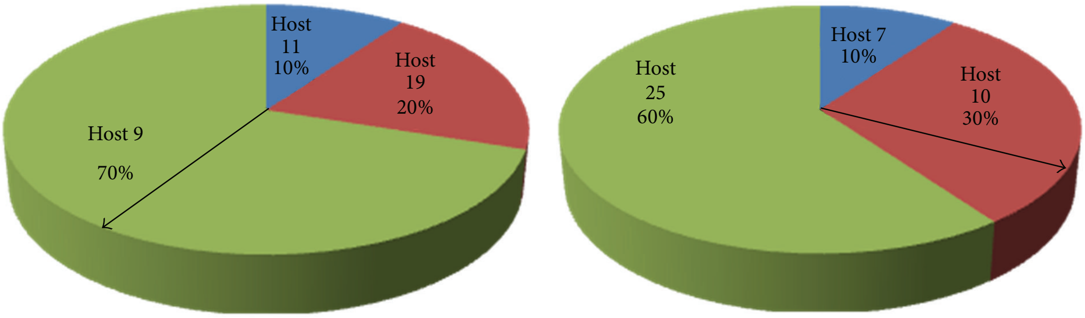
must be emphasized more than any other solution in the combined population. If the size of $S_1$ is larger than $N$, we utilized the $k$-means approach to cluster these individuals of $S_1$ for grouping the solutions into $N$ clusters. The $N$ clustering centers of the $N$ clusters are chosen to form the next population $P_1$. If the size of $S_1$ is smaller than $N$, we definitely choose all members of the set for the new population $P_1$. The remaining members of the population $P_1$ are chosen from subsequent nondominated fronts in the order of their ranking. Thus, solutions from the set $S_2$ are chosen next, followed by solutions from the set $S_3$, and so forth. This procedure is continued until no more sets can be accommodated. Assume that the set $S_l$ is the last nondominated set beyond which no other set can be accommodated. In general, the count of solutions in all sets from $S_1$ to $S_l$ would be larger than the population size. To choose exactly $N$ population members, we sort the solutions of the *last* front $S_l$ according to the crowded distance $D_p$ of the density estimation in descending order and choose the best solutions needed to make all population seats occupied.

The procedure of the policy which is used for forming the next population and ensures elitism of population evolution in MOGA-LS is shown in Figure 4. The new population $P_1$ of size $N$ is now used for selection, crossover, and mutation to create a new population $Q_1$ of size $N$. Repeat the above process until meeting the maximum number $i_{\max}$ of iterations. That is, the maximum generation number $i_{\max}$ of population evolution is reached. At that point, the final population obtained by evolving should be $P_{i_{\max}-1}$.

*Get the Final Solution Vector*. After the last round of iterations in MOGA-LS, the final population $P_{i_{\max}-1}$ is formed. The population $P_{i_{\max}-1}$ is sorted according to nondomination rank. Its first nondominated set ($\gamma_p = 1$) is the final Pareto optimal solutions set. Although the set of solutions vectors is the final solutions set of the multiobjective optimization problem proposed by us on the location selection policy of live VM migration, the migrant VMs cannot be migrated according to multiple solutions.

Generally, the way to address this problem is to choose a solution randomly or according to the context from the Pareto optimal set. However, as for the proposed Pareto multiobjective optimization problem of live VM migration policy in cloud environment, MOGA-LS is a heuristic approach which does not have and does not need any context information, and it should be a self-adaptive algorithm. Besides, randomly choosing a solution from the Pareto optimal set is nonefficient and one-sided.

In this paper, we have presented a novel approach to address the problem of selecting a solution from the obtained set of Pareto optimal solutions. Now, assume that there are $L$ solutions vectors in the Pareto optimal set. The $L$ solutions vectors can constitute a matrix $\Theta$ of $L * n$, where each row vector is a solution vector belonging to the Pareto optimal set. Each of the $n$ column vectors of the matrix $\Theta$ represents $L$ solutions of the location selection of the corresponding VM. For each of the $n$ migrant VMs, it has a solution space consisting of $L$ solutions. The $L$ solutions should consist of a few kinds of solutions. The probability of each solution is



(a) The probability wheel of VM 1

(b) The probability wheel of VM 2

Figure 5: Examples of the probability wheel.

obtained by calculating their frequency of occurrence in the $L$ solutions according to probability theory and mathematical statistics.

At this point, each migrant VM has a probability wheel as shown in Figure 5. A pointer randomly rotates on the probability wheel. The host in which the area where the pointer finally stops represents is the final target host of live migration. In the specific implementation, a random number limited between 0 and 1 can be used to find the final location of live VM migration by finding the range where the random number is. By utilizing this approach, each migrant VM gets the final solution of location selection of live VM migration. An $n$ dimension of solution vector is returned to migration controller as an optimal migration policy of the $n$ migrant VMs.

Let us consider a problem in the proposed approach. We know that the solution vector of the proposed problem ought to be a sequence of integer numbers which denote the hosts in the resource pool. However, each individual of the initial population is randomly initialized. What is more is that the crossover and mutation functions have some coefficients which are limited between 0 and 1. Thus, although the genes of each individual are limited to an integer type in the implementation, the problem that the solution is not an integer still persists. To address this problem, we do not limit the elements to an integer type but employ the smallest position value (SPV) rules presented by in [16, 17]. In short, it is a rule which converts each individual given by the GA to a valid solution vector fit for the proposed problem. The process of applying the SPV rules into the proposed approach can be understood as follows. First, the hosts in the resource pool should be numbered from 0 to $m-1$. Second, after the initial population is initialized, all the genes of each individual are sorted in ascending order and then are numbered from 0 to $n-1$ and have a modulo operation of $m$. For instance, in a time window $\Delta t$, there are four VMs to be migrated in a resource pool which has three hosts. If an individual is (−1.21, 3.29, −0.12, 1.26), it will be converted to (0, 3, 1, 2) firstly and then to (0, 0, 1, 2). At this point, the solution vector is useful and meaningful. It represents that the target host of VM 0 is host 0. The target host of VM 1 is host 0. The target host of VM 2 is host 1. The target host of VM 3 is host 2. In the MOGA-LS, all the solutions vectors (individuals) refer to the vectors which the original vectors have been converted to according to the SPV rules.

Starting from this intuition, the MOGA-LS algorithm which aims at live VM migration policy gives consideration to both the power saving optimization and the load balancing optimization to achieve a more efficient cloud data center by designing and utilizing the GA. During the course of the study, we have found that the two goals compete with each other to further improve itself. Thus, the GA cannot be utilized directly into our proposed problem. At this point, based on some inherent contradictory relationship between the two goals, we have thought of game theory, which is aiming to deal with and research this kind of problems. In the process that we further study game theory and multiobjective optimization, we have found the Pareto dominance theory. Based on the research on the Pareto dominance theory, we have proposed a novel multiobjective GA fit for being used to address the proposed problem and designed its genetic operators and fitness values as well as the policy of generating the next population. The specific process of MOGA-LS is also presented. In the MOGA-LS algorithm, each subalgorithm and subprocess are specially designed to better achieve our proposed two optimization goals. Therefore, MOGA is an obviously efficient heuristic algorithm for location selection policy of live VM migration from the experimental and design's points of view.

In the proposed MOGA-LS approach, we have introduced the simulated annealing (SA) idea [18] into the process of selecting the final solution from the obtained Pareto optimal set. Since the MOGA-LS algorithm is based on the Pareto dominance theory, the returned solution of MOGA-LS should be a set of Pareto optimal solutions (Pareto optimal front). In our problem scenario, it is very natural that only one final solution should be obtained as each VM needs to finally identify which host it will be migrated onto clearly. In the set of Pareto optimal solutions, it is obvious that all the solutions are equal and not comparable. Thus, the idea of comparing these solutions through other respects is meaningless and putting the cart before the horse. If so, we may as well directly add a new goal in the proposed MOP. Finally, the problem still persists. Thus, we think the way that a final solution is selected by comparing all the solutions of the Pareto optimal front does not work. The widely employed way is to randomly selecting a solution from the Pareto optimal front. Its rationality lies in that all the Pareto optimal solutions are optimal, equal, and not comparable as mentioned above. Although the method is simple and addresses this problem in a way, randomly choosing is only an expedient and obviously not efficient. Furthermore, our problem scenario is also not related to any context information to help MOGA-LS select the final solution, and the MOGA-LS approach is a self-adaptive heuristic algorithm, which needs to achieve the selection of the final solution autonomously. To address this problem in MOGA-LS, we have utilized the SA idea to convert the seeming disadvantage that multiple optimal solutions (Pareto optimal solutions set) are returned at last to the advantage that each VM finally obtains a sample space of optimal solutions and can be migrated to some optimal target host with a certain probability and thus to achieve a long-term optimization by implementing the SA idea. The specific process is shown above.

There are many parameters in the proposed MOGA-LS algorithm. Most of them have great influences on MOGA-LS. The maximum number $i_{\max}$ of iterations of the population evolution and the population size $N$ of MOGA-LS are two important parameters as well as their values are related to the efficiency and accuracy of MOGA-LS. For instance, if the population size $N$ increases, the computation time (convergence time) of MOGA-LS will increase, and the possibility that the algorithm is converged to the Pareto optimal front will also increase. That is, the global search ability will strengthen. Moreover, the required generation number $i_{\max}$ of population evolution will decrease with the population size $N$ increase. The maximum number $i_{\max}$ of iterations of the population evolution should be neither too small nor too large. Evidently, the size of the $i_{\max}$ value should make the first Pareto nondominated solutions set of the population of the MOGA-LS algorithm exactly converged to the Pareto optimal front as much as possible. Therefore, both the $i_{\max}$ value and the $N$ value are the experience problems and large numbers of experiments need to be performed to obtain fit values of them. In the proposed MOGA-LS approach, $N$ is set to 20 and $i_{\max}$ is set to 100 based on a large number of experiments. Also, the mutation rate $M_r$ is another important parameter in MOGA-LS since it has a direct impact on the diversity and convergence of population in MOGA-LS. It is difficult for GA to assign the mutation rate $M_r$ a fit value as it is related to a number of factors. Thus, it is an open problem which needs be studied further. In this paper, the $M_r$ value is set to 0.4. That is, there are 8 individuals to be mutated in each round of iterations in our proposed MOGA-LS approach.

## 4. Evaluation

In this section, we have experimentally verified the proposed MOGA-LS approach. The experiments have included evaluating the effect of power saving and load balancing of MOGA-LS and verifying the performance of MOGA-LS migration policy on the failure rate of migration events and SLA violation. Besides, an auxiliary experiment has been conducted to get a best power management policy in the cloud data center implementing the proposed MOGA-LS policy. In order to simulate a dynamic cloud data center, we have utilized an event driven simulator named CloudSim toolkit [19]. The CloudSim framework enables the accounting of the total power consumed by the system during the simulation period. In one word, the calculating of power consumption has been achieved by the CloudSim platform, which has provided a class including the methods getPower(). CloudSim allows such simulation scenarios by supporting dynamic creation of different kinds of entities and can add and remove data center entities at run time. This functionality has achieved simulating dynamic cloud environment where system components can join, fail, or leave the system randomly [19]. On CloudSim platform, we compare the proposed MOGA-LS approach with random migration policy and optimal migration policy by power consumption, the degree of load balancing and the number of invalid VM migration. We have prepared four different kinds of experiments to evaluate and test the proposed approach. The experimental results demonstrate that the proposed approach not only has a better power saving and load balancing but also has a better migration performance. Especially for a large number of migration events, the MOGA-LS approach shows the stability and the better performance.

*4.1. Experimental Scenarios.* On CloudSim platform, a resource pool consisting of 100 hosts is created. These hosts have varying computing resource. Twenty-four batches of virtual machine migration requests containing 13 requests randomly belonging to different hosts and with different resource requirements are created. The proposed MOGA-LS module is invoked and fetches the resource information and state of the cloud resource pool periodically. $\Delta t$ is set as 600 seconds, and the migration events of each batch are uniformly distributed within an hour, which includes $6\Delta t$ s. Besides, the host's resource change rate is set as 1 time per half an hour.

*4.2. Comparison in Power Saving.* The experiment is designed for verifying the efficiency and availability of MOGA-LS in power saving due to the location selection of live VM migration during the long-term operation of a cloud data center. In this scenario, we compare MOGA-LS with random migration policy, dynamic load balancing (DLB) policy implementing the idea of load balancing in [20], and the StdPSO migration policy which is an optimal migration policy based on standard particle swarm optimization for only power saving by power consumption during the long-term operation of the simulated cloud data center. As illustrated in Figure 6, the cloud data center implementing DLB migration policy and the cloud data center implementing random migration policy consume more power, while the cloud data center implementing MOGA-LS and the cloud data center implementing the StdPSO migration policies consume relatively less power. This is because both the latter two

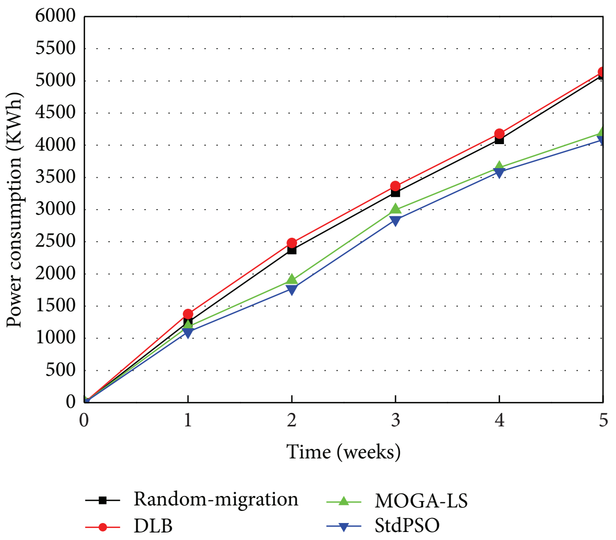


FIGURE 6: Comparison of random-migration, DLB, StdPSO, and MOGA-LS in power consumption.

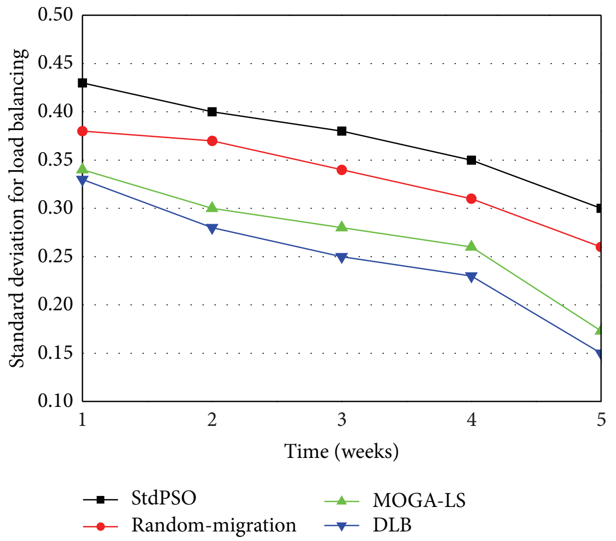


FIGURE 7: Comparison of StdPSO, random-migration, DLB, and MOGA-LS in load balancing.

are the heuristic approaches for power saving. Further, the cloud data center implementing DLB migration policy causes more power consumption than that of random migration policy. The reason for this is that the DLB migration policy has only taken load balancing into account. Achieving the balancing of load is to improve the service performance of cloud data center. What it has been considering is the user side. The migration policy for only the above goal naturally causes relatively more power consumption than the random migration policy since the two objectives, power consumption, and load balancing are competing with each other, and the random migration policy does not sway the balance on either side.

Besides, as can be seen in Figure 6, the cloud data center implementing the StdPSO migration policy has a less total incremental power consumption than the cloud data center implementing the MOGA-LS migration policy. The MOGA-LS approach has considered not only power saving but also load balancing. Just from the point of view on power-saving effect, the MOGA-LS policy is not better than the StdPSO policy as mentioned above. However, the power-saving effect of the MOGA-LS policy is very close to that of the StdPSO policy in Figure 6. The MOGA-LS algorithm has achieved many proposed optimization policy and thus had a better convergence to the Pareto optimal front and has utilized the SA idea to achieve a long-term optimization effect, effectively avoiding the local optimization. Therefore, although it has also considered load balancing, it obviously still has a better power-saving effect than the random migration policy and the DLB policy. Moreover, its power saving effect approximates that of the StdPSO algorithm for optimally saving power. To sum up, the proposed MOGA-LS approach is an efficient migration policy of live VM migration for power saving.

*4.3. Comparison in Load Balancing.* In the experiment scenario, it is designed for verifying the efficiency and availability of MOGA-LS in load balancing due to the location selection of live VM migration during the long-term operation of a cloud data center. We have compared the degrees of load balancing in the cloud data centers implementing the StdPSO policy, random migration policy, MOGA-LS migration policy, and DLB migration policy, respectively, at the end of each of five weeks. Here, we have employed the standard deviation value indicated above and used to measure the degree of load balancing to conduct the experiment. Obviously, a smaller standard deviation value represents that the cloud data center has the better balancing of load. We know that in a cloud data center there is the dedicated load balancing algorithm running regularly in order to actively achieve the balancing of system load. Thus, the degree of load balancing of a cloud data center will be getting better and better to some extent with the time passing. As shown in Figure 7, all the four policies' standard deviation values measuring the balancing of load have been tending to become smaller to some extent. However, at the end of each week, the obtained four policies' standard deviation values have shown that there are the certain gaps between the degrees of load balancing of cloud data centers. In Figure 7, the cloud data center implementing the StdPSO policy has the worst degree of load balancing. The reason is similar to the previous experiment. The cloud data center implementing the random migration policy has the second worst effect of load balancing, relatively, since it does not have any information. The DLB migration policy leads to the best effect of load balancing relatively. Meanwhile, the MOGA-LS approach is close to it and has the second best effect of load balancing. As can be seen in Figure 7, the four migration policies are divided into two levels of load balancing. In other words, the MOGA-LS migration policy and the DLB migration policy are the same level although the MOGA-LS algorithm is not as excellent as the DLB algorithm.

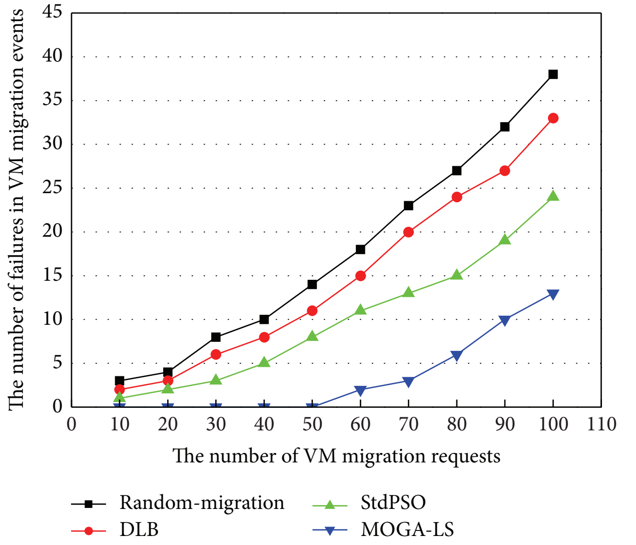


FIGURE 8: Comparison of the number of failures in VM migration events.

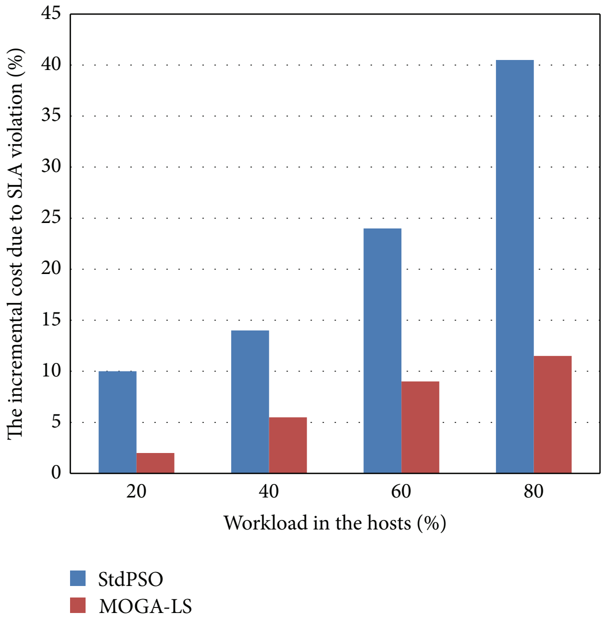


FIGURE 9: Comparison of the incremental cost due to SLA violation in VM migration with varying percentage of load in the cloud data center.

It is obvious that the MOGA-LS approach has already been quite efficient for load balancing. To sum up, the proposed MOGA-LS approach is an efficient and feasible migration policy of live VM migration for load balancing.

*4.4. Comparison of the Number of Failures in VM Migration Events.* In the experiment scenario, the dynamic host failure is simulated by CloudSim by scheduling some host failure events and host shut down events to occur during the location selection interval. These events have caused some failures in VM migration due to the nonavailability of the selected hosts for some VM migration requests. As illustrated in Figure 8, we have compared the MOGA-LS policy with the random migration policy, the DLB migration policy, and the mentioned StdPSO migration policy. In the simulated cloud data center, with the increase of the number of VM migration requests, the cloud data centers, respectively, implementing random migration, DLB and StdPSO result in more number of invalid VM migrations, whereas the MOGA-LS performs better in finding the fit hosts in the dynamic resource pool. This is because the memory data is outdated in the StdPSO, DLB, and random migration policy. They cannot have an adjustment with the environment changed. Conversely, since the SA idea has been introduced into it, the MOGA-LS policy based on the GA is quite efficient in detecting the host failures during the interval as well as has a fit adjustment in a better manner by searching out the new available hosts that can meet the resource requirements of the VM migration requests. The MOGA-LS approach can better meet the performance requirement of live VM migration and reduces the failure numbers of live VM migration since the MOGA-LS algorithm has seen the migration performance as the constraint objective. The design of the MOGA-LS approach is rational and efficient. After all, it is the migration policy of VMs that we are discussing. And it should first make live migration of VMs succeed as much as possible.

*4.5. Comparison of the Incremental Cost due to SLA Violation.* In this experimental scenario, we compare the incremental cost of StdPSO and MOGA-LS due to service level agreements (SLA) violation with varying percentage of load in the cloud data center. The load mentioned refers to the load of the whole cloud data center and is not the load of some physical host. As illustrated in Figure 9, with the increase of the percentage of workload in the cloud data center, the incremental cost of MOGA-LS is less than that of StdPSO due to SLA violation. We know if the workload in the hosts is heavier, achieving the balancing of load is more important for better service performance and to, thus, meet the SLA. The above experiment has shown that the StdPSO policy and the MOGA-LS policy belong to the same level for power saving. MOGA-LS is just slightly inferior than StdPSO. However, since the proposed MOGA-LS approach has also achieved the balancing of load, it has ability in making the cloud data centers increase resource utilization and provide better service performance to users. Thus, its cost due to SLA violation is less. So to say, the MOGA-LS approach has not only made a quite excellent power-saving effect come true to contribute to the green cloud data centers but also is considered the most important user experience and achieved load balancing and thus to enhance the resource utilization and service capability of green cloud data centers. To sum up, the MOGA-LS algorithm is an efficient location selection policy of live VM migration.

*4.6. Tradeoff between Power Saving and Performance Fulfillment.* The experimental scenario is conducted to find the optimal power management policy which balances the benefits due to power saving and performance fulfillment.

Since live VM migration events are time critical VM requests and the cloud service provider should meet strict SLA compliance, any violations in SLA in terms of performance loss of VMs will result in penalty cost on the provider of a cloud data center. The performance loss mentioned may be caused by power saving, live VM migration, network bandwidth, and throughput, and so forth. However, this paper has not focused on these problems. This experiment is designed to only find out a better power management policy suitable for the cloud environment implementing the proposed MOGA-LS to have a trade-off between the benefit of power saving and penalty cost due to SLA violation. In other words, the experiment aims to find a power management policy, which makes the proposed MOGA-LS approach more efficient in the cloud data centers and makes the cloud data centers implementing the MOGA-LS migration policy have a better power-saving effect and satisfy the users. There are four different policies to be formulated. The first policy is on/off policy, wherein all idle hosts are switched off. It can be seen that the policy gives the best power saving, but it causes the high penalty cost obviously. The single-DSS policy is the second policy, wherein all idle hosts are switched to deep sleep state. It results in an increase in the power consumption cost, but the penalty cost is reduced significantly. The third policy is single-SSS, wherein all idle hosts are switched to shallow sleep state. There is no penalty cost as SLA violation is absent in this policy. However, the enormous increase in the power consumption cost is caused. The multiple-SS is the fourth policy, wherein some of the idle hosts are kept in deep sleep, and others are kept in shadow sleep state according to a short-term prediction technology. From its meaning and as shown in Figure 10, it can be known that the multiple-SS policy gives the optimal cost trade-off, relatively.

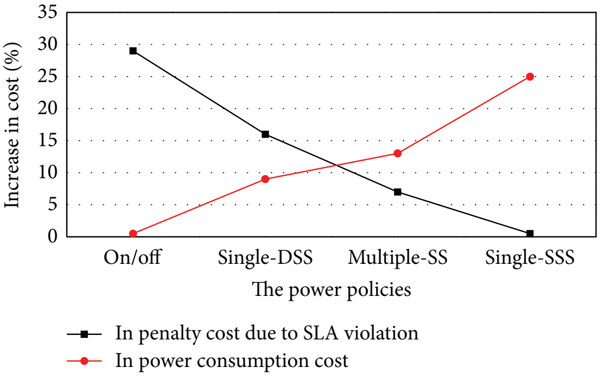


Figure 10: The trade-off between power and performance with different power management policies.

## 5. Conclusion and Future Work

In this paper, a novel location selection policy MOGA-LS of live VM migration is proposed, and we give its main idea, design, implementation, and evaluation. It employs the improved GA-based approach and the Pareto dominance idea. In the improved GA-based approach, we have designed the selection crossover, the crossover operator, and the mutation operator of MOGA-LS. Moreover, we have designed the fitness values of MOGA-LS and given how to obtain them. Also, in order to make the MOGA-LS approach have the elitism, we have designed and employed a novel $(\mu + \lambda)$ selection policy based on Pareto nondominated sorting to generate the next population and thus to further optimization the MOGA-LS algorithm.

It is noteworthy that in the proposed MOGA-LS approach we do not randomly get the initial population but employ a novel optimization method utilizing the hill-climbing technique and clustering technique. In this paper, we do not give its details. Besides, in the process of generating the next population, we have introduced $K$-means clustering into the process to address this problem that the number of the individuals of the first nondominated set may be larger than $N$. In the proposed MOGA-LS, we have also utilized the SA idea to obtain the final solution vector and achieve a long-term optimization. Also, aiming to connect the improved GA-based algorithm and the SA process, we take use of the probability theory and mathematical statistics as well as the characteristics (the returned solution is the set of Pareto optimal solutions) of the algorithm itself to obtain and process data.

MOGA-LS achieves the high-efficiency of power consumption and the stability of requirement performance. It not only minimizes the incremental power consumption of a cloud data center but also achieves the balancing of system load while minimizing the number of failure in VM migration events, relatively. In the proposed MOGA-LS approach, there are some open problems which need be studied further and experience problems which need many experiments to gradually get a better solution. The maximum number $i_{\max}$ of iterations of the population evolution and the population size $N$ of MOGA-LS are the experience problems and need to perform several experiments to obtain the fit values to, thus, make the MOGA-LS approach efficient and feasible. The mutation rate $M_r$ of our approach is an open problem which needs be further widely researched. In this paper, all the parameters are set to the fit values, respectively. To evaluate the MOGA-LS approach, we have conducted several experiments on the CloudSim platform. The final experimental results show that MOGA-LS has an excellent power-saving effect and achieves the balancing of system load while having a high success rate of live VM migration events. Therefore, MOGA-LS is an effective, heuristic, and self-adaptive location selection policy of live VM migration for power conservation and load balancing.

Aiming to further improve the performance of MOGA-LS, we plan to study the robustness of MOGA-LS in the next step work. MOGA-LS should have abilities in dealing with some sudden matters and be combined with the mechanism of live VM migration for achieving a more efficient hybrid management. In the next work and experiments, the experience problems and the open problems presented in this paper will also be researched further.

## Acknowledgments

The authors would like to thank the editors and anonymous reviewers for their valuable comments.

## References

[1] P. Barham, B. Dragovic, K. Fraser et al., “Xen and the art of virtualization,” in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP ’03)*, pp. 164–177, Bolton Landing, NY, USA, October 2003.

[2] Y. Li, W. Li, and C. Jiang, “A survey of virtual machine system: current technology and future trends,” in *Proceedings of the 3rd International Symposium on Electronic Commerce and Security (ISECS ’10)*, pp. 332–336, Guangzhou, China, July 2010.

[3] M. Armbrust, A. Fox, R. Griffith et al., “the clouds: a view of cloud computing,” Tech. Rep. EECS-2009-28, 2009.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: a tutorial,” *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[6] C. Rusu, A. Ferreira, C. Scordino, A. Watson, R. Melhem, and D. Mossé, “Energy-efficient real-time heterogeneous server clusters,” in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 418–428, San Jose, Calif, USA, April 2006.

[7] S. Srikantaiah, A. Kansal, and F. Zhao, “Energy aware consolidation for cloud computing,” *Cluster Computing*, vol. 12, pp. 1–15, 2009.

[8] A. Verma, P. Ahuja, and A. Neogi, “PMapper: power and migration cost aware application placement in virtualized systems (PMapper ’08),” in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, pp. 243–264, Springer, New York, NY, USA.

[9] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, “EnaCloud: an energy-saving application live placement approach for cloud computing environments,” in *IEEE International Conference on Cloud Computing (CLOUD ’09)*, pp. 17–24, Bangalore, India, September 2009.

[10] R. Jeyarani, N. Nagaveni, and R. V. Ram, “Self adaptive particle swarm optimization for efficient virtual machine provisioning in cloud,” *International Journal of Intelligent Information Technologies*, vol. 7, no. 2, pp. 25–44, 2011.

[11] S. Jing and K. She, “A novel model for load balancing in cloud data center,” *Journal of Convergence Information Technology*, vol. 6, no. 4, pp. 171–179, 2011.

[12] Z. Yang, G. Dong, and C. Li, “Algorithm study of load balance based on live migration of VM in cloud computing,” *International Journal of Advancements in Computing Technology*, vol. 5, no. 3, pp. 481–489, 2013.

[13] R. Jeyarani, R. V. Ram, and N. Nagaveni, “Implementation of efficient light weight internal scheduler for high throughput grid environment,” in *Proceedings of the National Conference on Advanced Computing in Computer Applications (NCACCA ’09)*, E. Jeyakumar and R. Rangarajan, Eds., pp. 283–289, Coimbatore, India, 2009.

[14] S. Sadhasivam, R. Jayarani, N. Nagaveni, and R. V. Ram, “Design and implementation of an efficient two-level scheduler for cloud computing environment,” in *Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing (ARTCOM ’09)*, pp. 884–886, Kottayam, India, October 2009.

[15] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, “A simulated annealing-based multiobjective optimization algorithm: AMOSA,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, 2008.

[16] M. F. Tasgetiren, M. Sevkli, Y. C. Liang, and G. Gencyilmaz, “Particle swarm optimization algorithm for permutation flowshop sequencing problem,” in *Ant Colony Optimization and Swarm*, vol. 3172 of *Lecture Notes in Computer Science*, pp. 382–389, Springer, Berlin, Germany, 2004.

[17] M. F. Tasgetiren, Y. C. Liang, M. Sevkli, and G. Gencyilmaz, “Particle swarm optimization algorithm for single machine total weighted tardiness problem,” in *Proceedings of the Congress on Evolutionary Computation (CEC ’04)*, pp. 1412–1419, Portland, Ore, USA, June 2004.

[18] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[19] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” *Software*, vol. 41, no. 1, pp. 23–50, 2011.

[20] M. H. Willebeek-LeMair and A. P. Reeves, “Strategies for dynamic load balancing on highly parallel computers,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 9, pp. 979–993, 1993.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2013, Article ID 652061, 6 pages
http://dx.doi.org/10.1155/2013/652061

*Research Article*

# Full Glowworm Swarm Optimization Algorithm for Whole-Set Orders Scheduling in Single Machine

**Zhang Yu**[1] **and Xiaomei Yang**[2]

[1] *Computer Science and Technology School, Taiyuan University of Science & Technology, Taiyuan 030024, China*
[2] *Economics and Management School, Taiyuan University of Science & Technology, Taiyuan 030024, China*

Correspondence should be addressed to Xiaomei Yang; yxm001_1@126.com

Received 7 July 2013; Accepted 1 September 2013

Academic Editors: Q. Cheng and J. Yang

By analyzing the characteristics of whole-set orders problem and combining the theory of glowworm swarm optimization, a new glowworm swarm optimization algorithm for scheduling is proposed. A new hybrid-encoding schema combining with two-dimensional encoding and random-key encoding is given. In order to enhance the capability of optimal searching and speed up the convergence rate, the dynamical changed step strategy is integrated into this algorithm. Furthermore, experimental results prove its feasibility and efficiency.

## 1. Introduction

Whole-set orders problem refers to customers' orders including multiple workpieces with different processing time and completion deadlines; since these workpieces are matching together as one, the delivery delay of the whole order will account for one delayed piece; customers are meeting a matching problem. In the customized production environment, whole-set orders problem can better reflect the corporations' service level and customers' satisfaction. It has become an important branch in the field of production scheduling and has broadened practical backgrounds.

At present, the algorithms used in production scheduling can be divided into accurate algorithms and approximation algorithms. Accurate algorithms (mathematical programming, branch and bound algorithm, Lagrangian relaxation, etc.) can get accurate solutions of problems, but big amount of calculation and time-consumption limit their applications in solving small-scale problems. Approximation algorithms (genetic algorithm, particle swarm optimization, ant colony algorithm, etc.) for their simple operation, parallel processing, have been widely applied in production scheduling and large-scale problems.

Glowworm swarm optimization (GSO) is proposed by Krishnanand and Ghose as one of the newest nature inspired heuristics [1], with it's simple model, less adjustable parameters, and fast convergence rate, which can be usually viewed in pattern recognition, routing, combinatorial optimization, and so forth [2–6]. In the optimizations of production scheduling, Kazem Sayadi et al. have successfully gotten the better solutions of permutation flow-shop scheduling problem [7]. Wu et al. have proved its feasibility and efficiency in the optimization of cross-dock scheduling [8]. Based on the discrete characteristic of whole-set orders and GSO's good performance in discretization, this paper presents an improved GSO for whole-set orders scheduling problem.

## 2. The Description of Weighted Whole-Set Orders Problem

*2.1. The Model of Whole-Set Orders Problem in Single Machine.* Due to the definition of whole-set orders problem, when each order includes only one workpiece, it becomes a problem of weighed number of delay jobs, so whole-set orders problem is a kind of NP-hard problem.

Maximizing the number of weighted whole-set orders is our objective function:

$$f = \max \sum_{h=1}^{H} w_h \ x_h. \tag{1}$$

We suppose the following.

(1) There are $N$ independent workpieces that need to be processed on one machine and belong to $H$ orders, $G_1, G_2, \ldots, G_H, G_h \ (1 \le h \le H)$ includes $j_1^h, j_2^h, \ldots, j_{n_h}^h$ jobs, $\sum_{h=1}^{H} n_h = N$, the weighted of $h$ is $w_h$.

Consider

$$\sum_{h=1}^{H} w_h = 1. \tag{2}$$

(1) The processing time of workpiece $j_j^h$ $(1 \le h \le H, 1 \le j \le n_h)$ is $p_j^h$ $(>0)$, and deadline is $d_j^h$.

(2) All the workpieces are got ready, which means arriving time $r_j^h = 0$.

(3) One piece can only be processed once, and single machine should process one job each time, $a_{ij}$ stands for if job $i$ was processed at the position of $j$, then $a_{ij} = 1$, else

$$a_{ij} = 0 \ , \quad \sum_{n=1}^{N} a_{ij} = 1, \quad j = 1, 2, \ldots, N. \tag{3}$$

(4) Once processed the workpiece should not be terminated.

(5) The complete time of workpiece $j_j^h$ is $C_j^h$, and it defines $x_h$ for whole-set coefficient as

$$x_h = \begin{cases} 1, & \text{if } \sum_{j=1}^{n_h} U_j^h = n_h, \\ 0, & \text{else,} \end{cases} \qquad U_j^h = \begin{cases} 1, & \text{if } C_j^h \le d_j^h, \\ 0, & \text{else.} \end{cases} \tag{4}$$

*2.2. Characteristics of Whole-Set Orders Problem.* The characteristics of whole orders problem include its complexity, restriction, and discreteness.

(1) *Complexity*. For a machining sort with $n$ workpieces, there may be $N$ factorial solutions. For example, if we get 7 customers, and 20 workpieces to machining, the total number of the solutions will be $2.4329e + 18$. This reflects that with the enlargement of the scheduling scale, the space of solutions will become lager, and the computation will increase exponentially. This needs to keep the diversity of metapopulation in solving whole-set orders problem, to shorten the solving time, to increase the probability of acquiring optimal solution and, to realize global optimization.

(2) *Restraintion*. As the optimal solution must meet the machine's or processing sequences' restraint conditions in whole-set orders problem, part of the sorts may become unfeasible scheduling solutions for not meeting the restraints. We should note metapopulation individual's validity in searching process when using glowworm swarm optimization and adopt revise strategies to unfeasible individual coming from location update to ensure the feasibility of the descendant.

(3) *Discreteness*. In classical GSO, the mobile step is usually a fixed numerical value. And this has good effect on solving continuous optimizing problems. But every metapopulation individual represents an independent panel point in whole-set orders problem, and unreasonable setting of the step may lead to mismatching situations in searching process. So in order to ensure the convergence effectiveness, we should do some dynamic handlings on step.

## 3. Glowworm Swarm Optimization for Whole-Set Orders Scheduling

*3.1. Description of Classical Glowworm Swarm Optimization.* Most kinds of glowworms can locate its position and exchange information by sending out rhythmed short beam. The idea of GSO is glowworm individual finding flaring neighbors in its searching scope. Move from initial position to a better one and at last assemble into one or more extreme value point.

In GSO algorithm, Glowworm individuals' attraction is only related to its brightness. Attraction of individual is proportional to brightness and inversely proportional to the distance between the two individuals. The position of individuals account for objective function value. Define dynamic decision domain as individual searching scope. When updating position, individuals move by step.

Detailed procedures of classical GSO are as follows.

(1) Initialize parameters. $n$ individuals are randomly placed in feasible region, $l_0$ accounts for fluorescein value, $r_0$ for dynamic decision domain, $s$ for step, $n_t$ for threshold in domain, $\rho$ for fluorescein elimination coefficient, $\gamma$ for fluorescein update coefficient, $\beta$ for update coefficient of domain, $r_s$ for maximal searching radius, and $t$ for iteration number.

(2) The objective function value $J(x_i(t))$ is transformed to $l_i(t)$ as

$$l_i(t) = (1 - \rho) l_i(t - 1) + \gamma J(x_i(t)), \tag{5}$$

in which $x_i(t)$ accounts for the position of individual $i$ at $t$ time.

(3) In each $r_d^i(t)$, select higher fluorescein value individuals forming a set of neighborhood $N_i(t)$. Hence,

$$N_i(t) = \left\{ j : \left\| x_j(t) - x_i(t) \right\| \le r_d^i(t) ; l_i(t) \le l_j(t) \right\}. \tag{6}$$

(4) The probability of individual $i$ may move toward $j$ as

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}, \tag{7}$$

in which $j$ is chosen by $p_{ij}(t)$.

(5) The position of individual $i$ can be updated as

$$x_i(t+1) = x_i(t) + s\left(\frac{x_j(t) - x_i(t)}{\left\| x_j(t) - x_i(t) \right\|}\right). \tag{8}$$

(6) The dynamic decision domain can be updated as

$$r_d^i(t+1) = \min\left\{r_s, \max\left\{0, r_d^i(t), \beta\left(n_t - \left|N_i(t)\right|\right)\right\}\right\}. \tag{9}$$

*3.2. Glowworm Swarm Optimization for whole-set orders Scheduling (GSOS).* By analyzing the characteristics of whole-set orders problem, this paper will solve it by using glowworm swarm optimization. The key of this algorithm includes encoding and decoding schema, individuals standing without neighbors, variable step strategy, and the distance of individuals.

*3.2.1. Encoding and Decoding Schema.* A sound encoding method can lower solving difficulty caused by constraint conditions and raise efficiency. The common encoding methods in approximation algorithms at present include machine-based coding and process-based coding. Among these methods, machine-based coding can reflect limitations of the processing machines, while it may easily result in deadlock scheduling; Jobs' relation-based coding uses binary coding method which may easily cause redundant scheduling results. Process-based coding assigns the same symbol to every process of the same job, but the result producing through this method is not an initiative scheduling.

In view of this, a hybrid encoding schema combining with 2-dimensional coding and random-key coding is given. In whole-set orders problem, one individual's location accounts for one feasible code, and the movement of individual means the exchange of codes.

In the coding, the method in this paper combines natural numbers and randomly real numbers between 0 and 1. Natural numbers 1,2,3,. . . $n$ stand for $n$ piece, and $x_i$ stands for positive real numbers between 0 and 1 without repetition generated randomly. As is shown in Table 1.

This method increases bits of valid number. It could avoid some repetitive sorts that may lead to invalid codes. In decoding, relative size of number accounts for the process position of workpiece $i$ in the sorts, ascending sort column $(x_1, x_2, ..., x_n)$, and the corresponding sort of natural numbers is the process sequence.

In example 1, there are 6 workpieces and their 2-dimensional codes like Table 2.

There are 6 workpieces and their 2-dimensional codes like Table 2.

Ascending sort the real numbers we get 0.01-0.23-0.44-0.48-0.76-0.79; the code of workpiece number 1 is 0.23, and 0.23 is the second minimum in the sort, so workpiece 1 will be processed at second position. Successively analyzing, we can get the process sequence: 4-1-5-2-3-6.

Table 1: 2-dimensional individual encoding.

| 1 | 2 | 3 | ⋯ | ⋯ | $n$ |
|---|---|---|---|---|---|
| $x_1$ | $x_2$ | $x_3$ | ⋯ | ⋯ | $x_n$ |

Table 2: Case of codes.

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| 0.23 | 0.48 | 0.76 | 0.01 | 0.44 | 0.79 |

Furthermore, all the feasible codes and process sequences are one-to-one correspondence. Because of the difference of each real number, their ascending sort would be uniqueness.

*3.2.2. Dispose of Individual without Neighbors.* According to classical GSO, an individual glowworm performs random move towards a neighbor better than itself through probability in accordance with the neighborhoods' fluorescent brightness. If there are too many candidate solution or the neighborhoods distribute unevenly, there may be a small probability of the event that some individuals do not have neighbors so that they would like to be stagnant, which may results in the slowing down of the rate of convergence and the likely access into the local optimization. To avoid the above disadvantages, every individual must be ensured to be dynamic in the optimizing process. Thus, in this algorithm, if an individual has no neighbor, it should move one step at random in the optimizing process of its own generation.

*3.2.3. Strategy of Improved Moving Step.* The step size of an individual in GSO is ordinarily fixed. In solving the whole-set orders problem or similar special problems, both bigger and smaller steps will cause adverse effects in that the bigger steps may result in missing the optimal solution, slowing down of the rate of convergence, and the easy occurrence of shaking, while the smaller steps may result in an early trapping in the local optimization by which the optimal solution cannot be obtained. Therefore, changing the step size dynamically can improve the solving efficiency of the algorithm.

At the beginning of the algorithm, the step size should be kept big enough to avoid trapping into the local optimization. With the increasing of iteration, the step size should be shortened gradually to ensure an optimal solution at the later stage of the algorithm. So the formula of moving step is designed as

$$\begin{gathered}(1-a) * st = s, \\ a = \exp\left(-20 \times \left(\frac{t}{T_{\max}}\right)^p\right),\end{gathered} \tag{10}$$

in which $s$ accounts for minimum step and can be initialized, $T_{\max}$ for maximum iteration, $t$ for current iteration, $p$ and for a integer between $[1, 30]$. Let $p = 5$; the curve of $a$ is shown as Figure 1.

```
(a) Initialization: $n, l_0, r_0, s, n_t, \rho, \gamma, \beta, r_s, T_{\max}, p$;
(b) While $(t \le T_{\max})$ then do:
        {
            for $i = 1$ to $n$; do:
                $l_i(t) = (1-\rho)\, l_i(t-1) + \gamma J(x_i(t))$;
                for each $i$ do:
                    { $N_i(t) = \{ j : \| x_j(t) - x_i(t) \| \le r_d^i(t); l_i(t) \le l_j(t) \}$;
                for each $j \in N_i(t)$; do:
                    { $p_{ij}(t) = \dfrac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}$;
                      $j = select(p_{ij})$;
                    $st = s / \left(1 - \exp\left(-20 \times (t/T_{\max})^p\right)\right)$;
                    $x_i(t+1) = x_i(t) + st \left( \dfrac{x_j(t) - x_i(t)}{\| x_j(t) - x_i(t) \|} \right)$;
                    $r_d^i(t+1) = \min\{ r_s, \max\{ 0, r_d^i(t), \beta(n_t - |N_i(t)|) \} \}$; }
                $t = t + 1$; }
```

Pseudocode 1

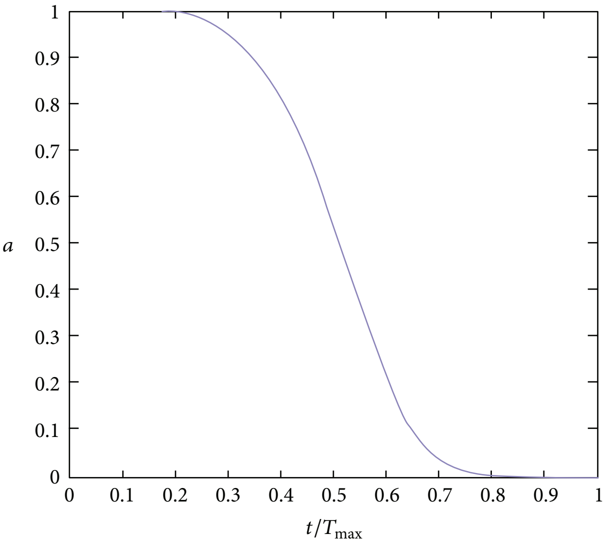


Figure 1: Curve of $a$.

Table 3: Orders and weights.

| Order | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Weight | 0.08 | 0.1 | 0.09 | 0.11 | 0.12 | 0.2 | 0.05 | 0.15 | 0.05 | 0.05 |

Table 4: Information of orders.

| | 2 | 3 | 7 | 10 | 14 | 16 | 19 | 30 | 38 | 42 | 52 | 60 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | (1) | | (2) | | | (2) | | (2) | | (2) | | (2) |
| 2 | (1) | | | (2) | (1) | | | | | | (2) | |
| 3 | | (1) | | | | | (2) | | (2) | | | (2) |
| 4 | | (1) | (2) | | (2) | | | (1) | | | (2) | |
| 5 | | (1) | | | | (1) | | | | (2) | | (2) |
| 6 | | | | (1) | | | (2) | | | | | (2) |
| 7 | | | | (1) | | (2) | | | (2) | | | |
| 8 | | | | (2) | (1) | | | (2) | | (1) | (2) | |
| 9 | | | | | (1) | | (1) | | (2) | | (1) | |
| 10 | | | | | (2) | | | (1) | | | | (2) |

*3.2.4. Formula of Distance between Individuals.* In whole-set orders problem, bit of codes is related to the numbers of workpiece and usually may be multidimensional array, and the key of it is how to calculate distance between individuals. Here, we suppose there are $N$ workpieces to be processed, and finally get two feasible sequences after decoding as follows:

$$x_i = (x_{i1}, x_{i2}, \dots, x_{iN}), \qquad x_j = (x_{j1}, x_{j2}, \dots, x_{jN}), \quad (11)$$

the distance from the symmetry of formula is obtained as

$$d_{ij} = \frac{|x_{i1} - x_{j1}|\, c + |x_{i2} - x_{j2}|\, c + \cdots + |x_{iN} - x_{jN}|\, c}{N(N+1)/2}, \quad (12)$$

in which $d_{ij} = d_{ji}$. Thus, $c$ is usually fixed, and we make it to be 4 in the experiments.

For example, there are two process sequences after decoding (6-5-4-3-2-1) and (1-2-4-3-6-5). By $d_{12} = 4 * \{(|6-1| + |5-2| + |4-4| + |3-3| + |2-6| + |1-5|)/(6*(6+1)/2)\} = 64/21$, the distance needed is obtained.

The pseudo codes of GSOS as shown in Pseudocode 1.

## 4. Simulations and Results

To verify the feasibility of GSOS, we have tested two simulations of different scales.

*Case 1.* According to [9], there are 41 workpieces of 10 orders, detailed information are shown in Tables 3 and 4; target of optimization is maximizing the number of weighted whole-set orders.

In Table 4, numbers in row-1 account for deadlines, in line-1 account for orders and in brackets for processing time.

Table 5: Comparison of time consumption.

| Algorithm | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Ave |
|---|---|---|---|---|---|---|---|---|---|---|---|
| GA | 121.8 | 117.9 | 118.2 | 117.6 | 118.2 | 117.5 | 119.6 | 117.9 | 117.8 | 118.6 | 118.5 |
| GSO | 84.90 | 83.70 | 83.81 | 83.15 | 83.39 | 83.72 | 84.19 | 83.63 | 83.54 | 83.52 | 83.76 |

Table 6: Results of numbers.

| Algorithm | Max | Min | Ave | Var |
|---|---|---|---|---|
| GA | 0.77 | 0.68 | 0.727 | 0.0762 |
| GSOS | 0.77 | 0.72 | 0.752 | 0.0531 |

Table 7: Orders and weights.

| Order | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Weight | 0.14 | 0.15 | 0.15 | 0.13 | 0.12 | 0.12 | 0.09 | 0.1 |

Table 8: (a) Information of orders for Job 1–10. (b) Information of orders for Job 11–21.

(a)

| Job | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Order | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 4 |
| Deadline | 8 | 23 | 40 | 10 | 26 | 42 | 15 | 32 | 48 | 17 |
| Cpu time | 4.5 | 4 | 3.5 | 4 | 2 | 4 | 3 | 3 | 3.5 | 3 |

(b)

| Job | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Order | 4 | 4 | 5 | 5 | 5 | 6 | 6 | 7 | 7 | 8 | 8 |
| Deadline | 36 | 56 | 20 | 39 | 60 | 12 | 35 | 16 | 40 | 22 | 50 |
| Cpu time | 3 | 5 | 3.5 | 4.5 | 3 | 2 | 4 | 4 | 4 | 3 | 2 |

Table 9: Results of numbers.

| Algorithm | Max | Ave | Var |
|---|---|---|---|
| GA | 0.77 | 0.728 | 0.0493 |
| GSOS | 0.78 | 0.749 | 0.0491 |

According to the above information, we contrast the results between GSOS and GA. Simulation environment: Microsoft Windows-XP system, AMD-A6-3400M CPU, 2G-RAM, and the codes are programmed by MATLAB2012a, the population size is fixed $n = 400$, maximum iteration $T_{\max} = 80$, and the program run 10 times independently.

The efficiency of results is shown in Table 5. From Table 5, we can find that average solving time of GSOS is shortened by about 34.74 seconds compared with GA, which increases by about 29.3%.

The results of optimization numbers are shown in Table 6. Form Table 6, we realize that GSOS performs better than GA in terms of average value, minimum value, and variance.

*Case 2.* To insure the performance of GSOS, Example 2 is generated randomly, information is in detail in Tables 7 and 8.

The results of optimization numbers are shown in Table 9, and the efficiency of results are shown in Table 10.

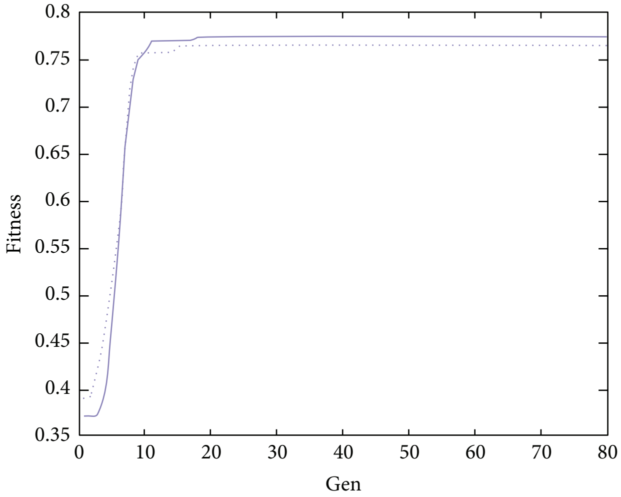


Figure 2: Curves in case 1.

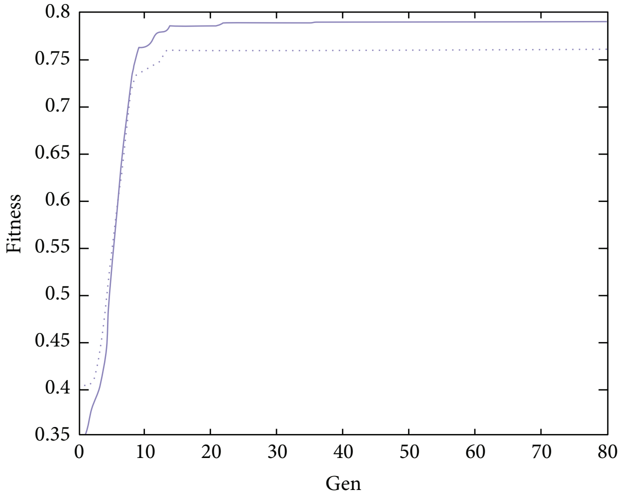


Figure 3: Curves in case 2.

From Tables 9 and 10, we can find that the proposed algorithm is better than GA.

The searching curves of GSOS and GA are shown in Figures 2 and 3, there solid lines account for GSOS while dotted lines for GA.

According to the Figures 2 and 3, GSOS and GA both have a high rate of convergence, but GSOS performs better than GA in terms of average value, minimum value, and variance, which proved its high-accuracy of solutions, the solving time of GSOS decreases by about 29% compared with GA which reflects its efficiency. In conclusion, GSOS is more suitable for solving whole-set orders problem.

Table 10: Comparison of time consumption.

| Algorithm | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Ave |
|---|---|---|---|---|---|---|---|---|---|---|---|
| GA | 121.8 | 117.9 | 118.2 | 117.6 | 118.2 | 117.5 | 119.6 | 117.9 | 117.8 | 118.6 | 118.5 |
| GSO | 84.90 | 83.70 | 83.81 | 83.15 | 83.39 | 83.72 | 84.19 | 83.63 | 83.54 | 83.52 | 83.76 |

## 5. Conclusions

An improved glowworm swarm optimization for scheduling (GSOS) is proposed in this paper; we have verified its high rate of convergence, efficiency, accuracy, and easy operation through simulations on different scales of whole-set orders problem. To test its performance on parallel machines and bigger scales that will be our research direction later on.

## Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grant no. 70971080) and the doctoral fund of Taiyuan University of Science & Technology (Grant no. 20102028). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## References

[1] K. N. Krishnanand and D. Ghose, "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 84–91, IEEE Press, June 2005.

[2] K. Deep and J. C. Bansal, "Mean particle swarm optimization for function optimization," *International Journal of Computational Intelligence Studies*, vol. 1, no. 1, pp. 72–92, 2009.

[3] K. N. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions," *Swarm Intelligence*, vol. 3, no. 2, pp. 87–124, 2009.

[4] S. Lukasik and S. Zak, "Firefly algorithm for continuous constrained optimization tasks," in *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, vol. 5796 of *Lecture Notes in Computer Science*, pp. 97–106, 2009.

[5] Y.-M. Li, Y. Q. Zhou, and X.-G. Yao, "Improved glowworm s-warm optimization based on the behavior of follow," *Computer Science*, vol. 38, no. 3, pp. 248–251, 2011.

[6] C.-P. Liu and C.-M. Ye, "Solving permutation flow-shop scheduling problem by firefly algorithm," *Industrial Engineering and Management*, vol. 17, no. 3, pp. 56–59, 2012.

[7] M. Kazem Sayadi, R. Ramezanian, and N. Ghaffari-Nasab, "A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems," *International Journal of Industrial Engineering Computations*, vol. 1, no. 1, pp. 1–10, 2010.

[8] B. Wu, C.-H. Qian, and W.-H. Ni, "Glowworm swarm optimization for cross dock scheduling problem," *Computer Engineering and Applications*, vol. 30, no. 1, pp. 155–161, 2013.

[9] S.-Y. Zhou and R.-Q. Chen, "Genetic algorithm for maximizing the weig-hted number of whole-set orders," *Systems Engineering*, vol. 23, no. 5, pp. 22–24, 2005.