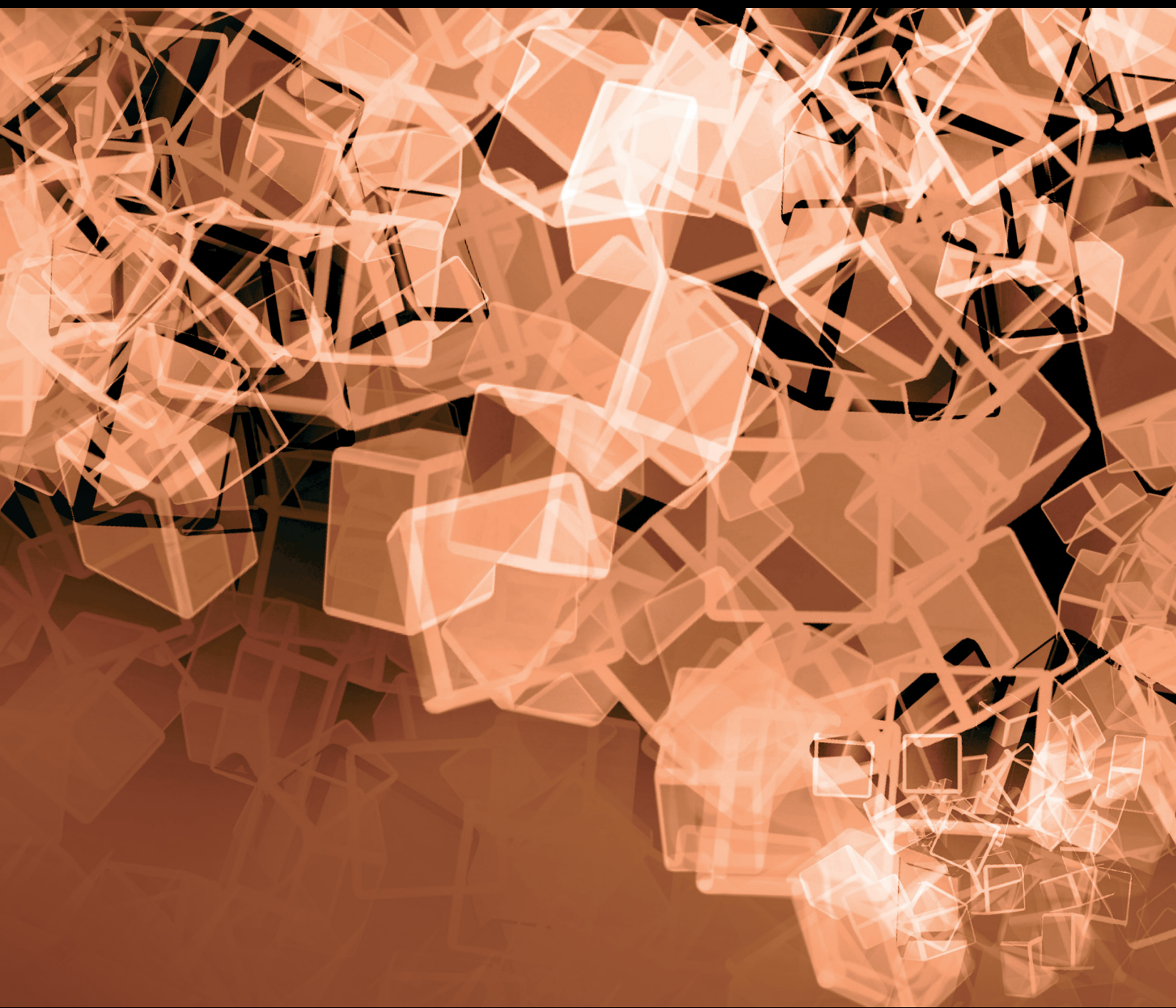
Discrete Dynamics in Nature and Society

Discrete Dynamics in Evolutionary Computation and Its Applications

Guest Editors: Yong-Hyuk Kim, Ahmed Kattan, Michael Kampouridis, and Yourim Yoon



Discrete Dynamics in Evolutionary Computation and Its Applications

Discrete Dynamics in Nature and Society

Discrete Dynamics in Evolutionary Computation and Its Applications

Guest Editors: Yong-Hyuk Kim, Ahmed Kattan, Michael Kampouridis, and Yourim Yoon

Editorial Board

Contents

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2016, Article ID 6043597, 2 pages
http://dx.doi.org/10.1155/2016/6043597

Editorial

Discrete Dynamics in Evolutionary Computation and Its Applications

Yong-Hyuk Kim,[1] Ahmed Kattan,[2] Michael Kampouridis,[3] and Yourim Yoon[4]

[1]*Department of Computer Science and Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01890, Republic of Korea*
[2]*EvoSys Ltd., King's Road Tower, Level 26, Jeddah 21499, Saudi Arabia*
[3]*School of Computing, University of Kent, Medway Campus, Chatham Maritime, Kent ME4 4AG, UK*
[4]*Department of Computer Engineering, Gachon University, 1342 Sengnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do 13120, Republic of Korea*

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

Received 31 May 2016; Accepted 31 May 2016

Evolutionary computation (EC) is considered to be a natural and artificial system with discrete dynamics. Over the last few decades, there has been a remarkable growth in the field of EC, encompassing large interest and efforts from researchers. EC has been successfully applied to various real-world problems for optimization purposes. The aim of this special issue is to publish original and high-quality articles related to discrete dynamics in EC and its applications.

This special issue was opened in November of 2015 and closed in February of 2016. There were a total of 29 submissions. All of them were peer-reviewed according to the high standards of this journal and only 5 of them were accepted for publication, which gave important developments in discrete dynamics in EC and its applications. Among the accepted papers, one is for the discrete dynamics in EC and the others are for two major applications of EC, operations research and network problems. The guest editors of this special issue hope that the presented results could outline new ideas for further studies.

In EC, selection or mating is one of the most important operations. In the paper entitled "A New Adaptive Hungarian Mating Scheme in Genetic Algorithms," C. Jung et al. suggested an adaptive mating scheme from Hungarian mating schemes, which consist of maximizing the sum of mating distances, minimizing the sum, and random matching. They presented an adaptive algorithm to elect one of these Hungarian mating schemes. Each mated pair of individuals voted for the next generation mating scheme. The distance between parents and the distance between parent and offspring were considered during voting. Two well-known combinatorial optimization problems, the traveling salesman problem and the graph bisection problem, which are NP-hard, were considered to show the effectiveness of their adaptive method.

In the paper entitled "A Hybrid IP/GA Approach to the Parallel Production Lines Scheduling Problem," H. Ren and S. Sun studied a scheduling problem of parallel production lines. Considering the time window and some technical constraints, they formulated a mixed integer programming model for the problem. They also deduced some valid inequalities and presented a hybrid mixed integer programming/constraint programming decomposition strategy. Based on them, the authors proposed a hybrid genetic algorithm for efficiently solving the problem.

In the paper entitled "Solving a Closed-Loop Location-Inventory-Routing Problem with Mixed Quality Defects Returns in E-Commerce by Hybrid Ant Colony Optimization Algorithm," S. Deng et al. presented a closed-loop location-inventory-routing problem model considering both quality defect returns and nondefect ones in e-commerce supply chain system, to minimize the total cost produced in both forward and reverse logistics networks. They proposed a hybrid ant colony optimization algorithm for efficiently solving this model that is NP-hard.

Since various factors affect the fluctuation of network traffic, accurate prediction of network traffic is considered as a challenging task of the time series prediction process. In the paper entitled "A Network Traffic Prediction Model Based on Quantum-Behaved Particle Swarm Optimization Algorithm and Fuzzy Wavelet Neural Network," K. Zhang et al. proposed a novel prediction method of network traffic based on quantum-behaved particle swarm optimization (QPSO) algorithm and fuzzy wavelet neural network (FWNN). The authors introduced QPSO and presented the structure and operation algorithms of FWNN. The parameters of FWNN were optimized by a QPSO algorithm. This optimized QPSO-FWNN was applied to the prediction of network traffic successfully when compared to different prediction models such as BP neural network, RBF neural network, fuzzy neural network, and FWNN-GA neural network.

The exponential growth in data traffic due to the modernization of smart devices has resulted in the need for a high-capacity wireless network in the future. To successfully deploy 5G networks, we should be able to handle the growth in the data traffic. The increasing amount of traffic volume puts excessive stress on the important factors of the resource allocation methods such as scalability and throughput. In the paper entitled "A Genetic Algorithm with Location Intelligence Method for Energy Optimization in 5G Wireless Networks," R. Sachan et al. defined network planning as an optimization problem with the decision variables such as transmission power and transmitter location in 5G networks, leading to interesting implementation using some heuristic approaches such as differential evolution and real-coded genetic algorithm (RCGA). The authors modified an RCGA-based method to find the optimal configuration of transmitters by not only offering optimal coverage of underutilized transmitters, but also optimizing the amounts of power consumption.

In sum, the five papers present some of the latest and most promising research results on EC and its applications. This special issue demonstrates the theoretical and practical importance of further studies on EC. The guest editors of this special issue hope that these papers may enrich and provide a guide to the readers to treat EC or further developments in the applications of EC.

Acknowledgments

The guest editors of this special issue would like to express their gratitude to all of the authors for their contributions and the reviewers for their effort providing helpful comments and suggestions. This work was supported by the Gachon University Research Fund of 2015 (GCU-2015-0030) and a grant [MPSS-CG-2016-05] through the Disaster and Safety Management Institute funded by the Ministry of Public Safety and Security of the Korean government.

Yong-Hyuk Kim
Ahmed Kattan
Michael Kampouridis
Yourim Yoon

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2016, Article ID 6467812, 12 pages
http://dx.doi.org/10.1155/2016/6467812

Research Article

Solving a Closed-Loop Location-Inventory-Routing Problem with Mixed Quality Defects Returns in E-Commerce by Hybrid Ant Colony Optimization Algorithm

Shuai Deng,[1,2] Yanhui Li,[1] Hao Guo,[1] and Bailing Liu[1]

[1] *School of Information Management, Central China Normal University, Wuhan 430079, China*
[2] *School of Economic Management, Hunan Institute of Technology, Hengyang 421001, China*

Correspondence should be addressed to Hao Guo; haoguo@mails.ccnu.edu.cn

Received 27 November 2015; Revised 9 March 2016; Accepted 19 May 2016

Academic Editor: Yourim Yoon

This paper presents a closed-loop location-inventory-routing problem model considering both quality defect returns and nondefect returns in e-commerce supply chain system. The objective is to minimize the total cost produced in both forward and reverse logistics networks. We propose a combined optimization algorithm named hybrid ant colony optimization algorithm (HACO) to address this model that is an NP-hard problem. Our experimental results show that the proposed HACO is considerably efficient and effective in solving this model.

1. Introduction

According to eMarketer, worldwide business-to-consumer (B2C) e-commerce sales reached $1.471 trillion in 2014, increasing by nearly 20% over 2013 [1]. Customers have grown accustomed to return unwanted products back to the store for any reasons. It is reported that the proportion of customer returns online range from 18% to 74% of original orders under e-commerce environment [2, 3]; thus enterprises take various measures to prevent the appearance of quality defects. However, quality defect is inevitable. So, it is necessary for us to take into account both quality defect returns and nondefect returns; we call it mixed quality defect returns (MQDR), when considering the closed-loop supply chain as a support system in e-commerce environment.

As a classic discrete dynamics problem, the customer service level is determined by three important decisions: facility location decision, inventory decision, and transportation decision [4]. Obviously, facility location, inventory control, and transportation optimization are highly related. For example, delivery in small lots and high frequency leads to reducing the in-inventory cost but increases the additional transportation cost. In addition, facility location decision needs to consider inventory decision and distribution decision. Perl and Sirisoponsilp [5] discuss the interdependence between the three key elements. Ballou and Masters [6] provide a schematic representation of the interrelationships among facility location, inventory control, and transportation optimization.

In the literature, many papers studied the integration and coordination of any two of the above three decisions: location-inventory problem (LIP), location-routing problem (LRP), and inventory-routing problem (IRP). For reviews on LIP, readers can refer to Erlebacher and Meller [7], Daskin et al. [8, 9], and Liao et al. [10]. For LRP, please refer to Balakrishnan et al. [11], Min et al. [12], and Nagy and Salhi [13]. Refer to Chan et al. [14], Kleywegt et al. [15], and Adelman [16] for IRP.

There are few researches about the integration optimization of location-inventory-routing problem (LIRP). Some researchers attempt to carry out research on LIRP [17]. Liu and Lee [18] firstly studied this interesting problem; they proposed a two-phase heuristic method to solve the multidepot location-routing problem (MDLRP) considering inventory optimization. In order to avoid the local optimal solution, Liu and Lin [19] designed a global optimizing

heuristic method to find the solutions for LIRP. Shen and Qi [20] presented an algorithm based on Lagrangian relaxation to minimize the inventory and routing costs in strategic location models. They focused on the layout phase and used continuous approximation to get the approximate optimal routing cost, but the vehicle routing was not optimized in their models. Javid and Azad [21] presented a novel LIRP model and proposed heuristic method containing two stages: constructive stage and improvement stage. Ahmadi-Javid and Seddighi [22] presented a mixed-integer programming model and a three-phase heuristic to solve the LIRP with multisource distribution logistics network. Guerrero et al. [23] researched the LIRP with deterministic demand and provided the hybrid algorithm to solve the problem. Zhang et al. [24] proposed a hybrid metaheuristic solution to LIRP considering multiple depots and geographically dispersed customers. Nekooghadirli et al. [25] presented a novel biobjective model of LIRP model considering a multiperiod and multiproduct system. Based on Lagrangian relaxation and a column generation technique, Guerrero et al. [26] developed a relax-and-price heuristic to solve ILRP; they proposed two dependent constraint sets with an exponential nature: Lagrangian relaxation and a column generation technique.

However, little research has been conducted on the LIRP considering returns. Li et al. [27] presented the HGSAA algorithm to solve a LIRP model considering returns under e-supply chain environment. To be more consistent with reality, Liu et al. [28] introduced a stochastic demand into LIRP considering returns in e-commerce and proposed a PPGASA algorithm as the solving approach.

The above two researches mainly focus on the returns without quality defect but did not consider the MQDR. In this paper, we propose a model of closed-loop LIRP with MQDR. To the best of our knowledge, it is the first time to introduce the MQDR into LIRP in e-commerce. An effective hybrid algorithm named hybrid ant colony optimization (HACO) is provided to solve this model. Results of numerical instances indicate that HACO outperforms ant colony optimization (ACO) on optimal solution, iterations, and computing stability.

The remainder of this paper is organized as follows. Section 2 presents the mathematical model of LIRP with MQDR. Section 3 proposes the solution approach named HACO. Section 4 analyses the parameters of HACO and shows the results of different experiments. Section 5 gives the conclusion and future research directions.

2. Model Formulation

As we all know, customers' return in e-commerce is higher than traditional commerce. Because of personal dissatisfaction, or a mistaken purchase of the wrong product, some of the returns are without quality defects. These returns can reenter into the market after a simple repackaging process without being recovered [29]. While the other returns result from quality defects, which need to be sent back to the plant and be recovered.

In order to meet the needs of MQDR, the merchandise center (MC) is necessary to deliver normal merchandises

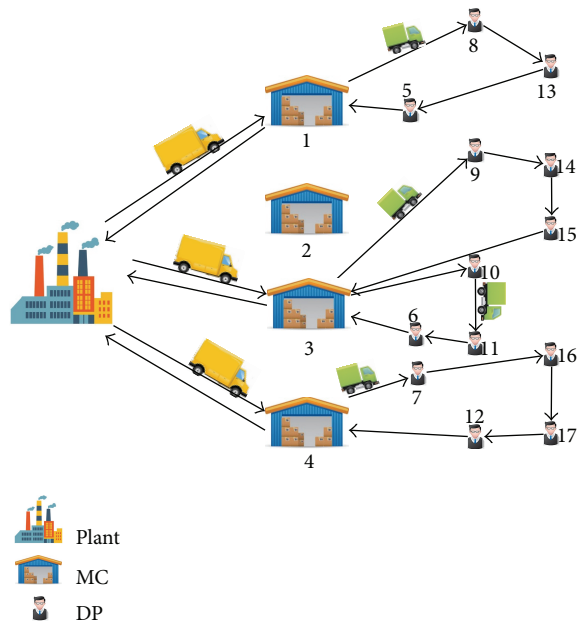


Figure 1: Closed-loop supply chain for a single product.

to the demand points (DPs) of downstream and collect the returned merchandises. MC integrates the functions of distribution center and recycling center and provides quality inspection and repackaging services. Meanwhile the returned merchandises are collected to MCs. Returned merchandises without quality defects become resalable normal items after repackaging treatment at MCs. The plant will recover the returns with quality defects and bring them to the market again.

The operation mode of the system is shown in Figure 1. The closed-loop supply chain in this paper consists of one plant, multiple MCs, multiple DPs, and a single type of product with continuous inventory policy under the e-commerce environment.

The goal of this study is to decide the quantity and location of MCs and arrange the vehicle routes and determine the ordering times on each route. To minimize the total cost of logistics operations, this problem involves the following three decisions: (1) location decisions: obtain the optimal number of MCs and their locations; (2) inventory management: determine the ordering times on each route; (3) routing optimization: arrange the vehicles to delivery merchandises and collect returns.

To benefit from the risk of MQDR, we take assumptions (1)–(8) from Li et al. [27] into consideration: since the single-product system is researched in this paper, assumption (1) is necessary; in the capacitated vehicle routing problem, assumption (2) should be satisfied [30]; assumption (3) eliminates the indeterminacy from the different type of vehicle; assumption (4) means that each DP is well served by the only vehicle route [31]; assumption (5) ensures that each route will return to the same MC after traversing; assumption (6) follows the early published papers considering uncapacitated MCs [32]; assumption (7) takes MCs as the distribution

center and recycling center; assumption (8) is a simplification of the reality [33].

The returned merchandises without quality defect are processed and repackaged at MCs, while others will be shipped back to the plant for reprocessing after a predetermined quantity at the MCs. Assume that the demand at each retailer is known and let **R** be the set of candidate MCs. Let **S** be the set of DPs and let M be the number of DPs. Let **V** be set of vehicles from the MCs to DPs. Let $\mathbf{U} = \mathbf{R} \cup \mathbf{S}$. Let the following notation denote the decisions of the firm:

N_{rv}: ordering times of MC_r on routing v.

X_{ijr}^{v}: =1, if node j is served by MC_r on routing v from node i, or 0 otherwise.

Y_{ir}^{v}: =1, if node i is assigned to MC_r on routing v, or 0 otherwise.

Z_r: =1, if MC_r is selected as an MC location, or 0 otherwise.

U_{jk}: auxiliary variable avoiding the subtour constraints in route k.

According to the aforementioned assumptions, the inventory levels depend on both demand and the quantity of MQDR. So, during each replenishment cycle, the holding cost of MCs is $\lambda h \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} ((d_i + q_i + w_i)/2N_{rv}) Y_{ir}^{v}$, where h is annual inventory holding per unit merchandises, d_i is mean (daily) demand for DP_i, and q_i and w_i are quantity of merchandises without and with quality defect returned by DP_i per day.

In order to exactly describe the logistic distribution costs. Let c_r be the transportation costs per unit product from plan to MC_r. Let l be the delivering cost per unit distance. Let s_{ij} be the distance from node i to node j. And let λ be the working days per year. The total transportation costs from plant to DPs through MCs can be expressed, respectively, as $\lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} c_r (d_i - q_i + w_i) Y_{ir}^{v}$ and $\lambda l \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} \sum_{j\in\mathbf{U}} s_{ij} d_i X_{ijr}^{v}$.

So the cost of forward distribution is

$$\begin{aligned} &\sum_{r\in\mathbf{R}} f_r Z_r + \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} b_r N_{rv} + \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} e_r N_{rv} \\ &+ \lambda h \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} \frac{(d_i + q_i + w_i)}{2N_{rv}} Y_{ir}^{v} \\ &+ \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} c_r (d_i - q_i + w_i) Y_{ir}^{v} \\ &+ \lambda l \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} \sum_{j\in\mathbf{U}} s_{ij} d_i X_{ijr}^{v}, \end{aligned} \tag{1}$$

where e_r denote the fixed cost of dispatching vehicles per time at MC_r, f_r denote the fixed (annual) administrative and construction cost of MC_r, and b_r denote the ordering cost per unit product from plant to MC_r.

We let k to be the returning cost per unit of merchandise from DPs to MCs, so the total reverse transportation costs from DPs back to MCs are

$$\lambda k \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} (q_i + w_i) Y_{ir}^{v}. \tag{2}$$

The cost of deal with mixed quality defects is

$$\lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} p_r q_i Y_{ir}^{v} + \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} a_r (q_i + w_i) Y_{ir}^{v}. \tag{3}$$

We adopt a_r as the inspecting cost per unit for the returned product and p_r as the repackaging cost of unit returned merchandise without quality problem at MCs.

In summary, the model is formulated as follows:

$$\begin{aligned} \min Z = &\sum_{r\in\mathbf{R}} f_r Z_r + \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} b_r N_{rv} + \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} e_r N_{rv} \\ &+ \lambda h \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} \frac{(d_i + q_i + w_i)}{2N_{rv}} Y_{ir}^{v} \\ &+ \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} c_r (d_i - q_i + w_i) Y_{ir}^{v} \\ &+ \lambda l \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} \sum_{j\in\mathbf{U}} s_{ij} d_i X_{ijr}^{v} \\ &+ \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} p_r q_i Y_{ir}^{v} \\ &+ \lambda k \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} (q_i + w_i) Y_{ir}^{v} \\ &+ \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} a_r (q_i + w_i) s Y_{ir}^{v}. \end{aligned} \tag{4}$$

It is easy to find that the objective function (4) is convex in N_{rv}. We can simplify the objective function by solving N_{rv}. Consequently, the optimization solution of N_{rv} can be obtained by taking the derivative of the function with respect to N_{rv}; the result is as follows:

$$N_{rv}^{*} = \sqrt{\frac{\sum_{i\in\mathbf{S}} \lambda h (d_i + q_i + w_i) Y_{ir}^{v}}{2(e_r + f_r)}}. \tag{5}$$

The optimization problem (4) given a known N_{rv}^{*} can now be written as

$$\begin{aligned} \min Z = &\sqrt{2\lambda h \sum_{r\in\mathbf{R}} \sum_{v\in\mathbf{V}} \sum_{i\in\mathbf{S}} (f_r + e_r) (d_i + q_i + w_i) Y_{ir}^{v}} \\ &+ \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} c_r (d_i - q_i + w_i) Y_{ir}^{v} \\ &+ \lambda l \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{U}} \sum_{j\in\mathbf{U}} s_{ij} d_i X_{ijr}^{v} \\ &+ \lambda k \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} (q_i + w_i) Y_{ir}^{v} \\ &+ \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} p_r q_i Y_{ir}^{v} \\ &+ \lambda \sum_{v\in\mathbf{V}} \sum_{r\in\mathbf{R}} \sum_{i\in\mathbf{S}} a_r (q_i + w_i) Y_{ir}^{v} + \sum_{r\in\mathbf{R}} f_r Z_r \end{aligned} \tag{6}$$

$$\text{s.t.} \quad \sum_{r\in\mathbf{R}} Z_r \geq 1, \quad r \in \mathbf{R} \tag{7}$$

$$\sum_{v\in\mathbf{V}}\sum_{i\in\mathbf{S}} Y_{ir}^{v} \leq Z_r, \quad r \in \mathbf{R} \tag{8}$$

$$\sum_{i\in\mathbf{S}} d_i Y_{ir}^{v} \leq g, \quad r \in \mathbf{R},\ v \in \mathbf{V} \tag{9}$$

$$\sum_{v\in\mathbf{V}}\sum_{r\in\mathbf{R}} Y_{ir}^{v} = 1, \quad i \in \mathbf{S} \tag{10}$$

$$\sum_{v\in\mathbf{V}}\sum_{r\in\mathbf{R}}\sum_{i\in\mathbf{U}} X_{ijr}^{v} = 1, \quad j \in \mathbf{S} \tag{11}$$

$$\sum_{i\in\mathbf{U}} X_{ikr}^{v} - \sum_{j\in\mathbf{U}} X_{kjr}^{v} = 0, \quad k \in \mathbf{S},\ r \in \mathbf{R},\ v \in \mathbf{V} \tag{12}$$

$$U_{lk} - U_{jk} + Mx_{ljk} \leq M - 1, \quad l, j \in \mathbf{S},\ k \in \mathbf{R} \tag{13}$$

$$-Y_{ij}^{v} + \sum_{k\in\mathbf{U}} \left(X_{ikr}^{v} + X_{kjr}^{v}\right) \leq 1, \quad i \in \mathbf{S},\ j, r \in \mathbf{R},\ v \in \mathbf{V} \tag{14}$$

$$Z_r = \{0, 1\}, \quad r \in \mathbf{R} \tag{15}$$

$$Y_{ir}^{v} = \{0, 1\}, \quad i \in \mathbf{S},\ r \in \mathbf{R},\ v \in \mathbf{V} \tag{16}$$

$$X_{ijr}^{v} = \{0, 1\}, \quad i \in \mathbf{U},\ j \in \mathbf{S},\ r \in \mathbf{R},\ v \in \mathbf{V}. \tag{17}$$

The objective function (6) is to minimize the total cost; (7) ensure the selected MC is not empty; (8) ensure each DP is traversed by a unique vehicle which belongs to a certain MC; (9) ensure the amount of each delivery from MC on each route must be within vehicle capacity; (10) ensure that each route has only one vehicle; (11) ensure each DP must be followed by exactly one note; (12) ensure every DP node of the system will be serviced before it gives services to the others; (13) ensure the subtour is eliminated. Equation (14) ensures that each DP is assigned to an MC when there exists a route that starts from the MC passing through the DP. Equations (15)–(17) ensure the nonnegativity and integrality of decision variables.

3. Solution Approach

Like the VRP, the closed-loop LIRP is also an NP-hard problem, since it includes the VRP and is more complex than VRP. Generally speaking, there does not exist a complete, efficient, and accurate analytic algorithm to address NP-hard problems; ant colony optimization (ACO) has been proved very successful and widely applied to solve the static and dynamic problems as an EC algorithm [34]. However, ACO does not distinguish ant behavior results. The pheromone concentration will distribute in every direction for the iteration. Therefore, this leads to low searching efficiency. The algorithm may get caught in local optimization if we do not take preventive measures. On the other hand, ABC provides an effective institution to find the global optimal solution from the trapping of local optimal solution [35]. So, in this study, we present a hybrid ant colony optimization algorithm based on the combination of ACO and ABC to solve the above LIRP model.

3.1. Initialize Solution. Since the natural number is an efficient coding method for these problems, the sequence of solutions is composed of candidate MCs $(1, 2, \ldots, R)$ and DPs, which are indicated by $(R + 1, \ldots, R + S)$. The candidate solution of our proposed model will be described well by those natural number sequences. As an example, Figure 1 fully interpreted the perceptions of our method, which refers to the individual feasible solution: {1 8 13 5 3 9 14 15 10 11 6 4 7 16 17 12}.

In the HACO, the moving strategy of the ant in node i is depending on the pseudorandom proportional rule. The rule indicates that the ant has both exploiting and exploration ability, which means the ant is guided by the pheromone trails as well as the heuristic information. In this case, the ant has a higher degree of exploring unknown knowledge. The connected function of the pheromone values τ_{ij} and the heuristic values η_{ij} is shown as

$$P_{ij}^{k} = \begin{cases} \dfrac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{k\in\text{allow}(k)}\left[\tau_{ik}\right]^{\alpha}\left[\eta_{ik}\right]^{\beta}} & j \in \text{allow}\,(k) \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where τ_{ij} is the density of pheromone remaining on the edge (i, j), η_{ij} is the inverse value of distance between node i and node j, α and β are user-defined parameters for corresponding pheromone concentration and heuristic information, and allow(k) is the remaining nodes to be visited by ant k.

3.2. The ABC Phase. In order to improve the performance of global searching of our algorithm, the paper applied the scout bee searching phase into the ACO. Scout bees are free bees used for finding a new better solution from the neighbor known solution. As soon as a scout bee finds a new solution, she turns into an employed bee. If there is no improvement in the quality of solution, the bee will abandon that source and continue to search for another new solution.

The searching function of scout bees is as

$$\widehat{x}_i^{j} = x_{\max} - \text{rand}\,[0, 1] * \left(x_{\max} - x_i^{j}\right). \tag{19}$$

To meet the requirements for coding sequence type, we described two operations to complete scouts searching process, namely, random array reverse (RAR) and random swap (RS).

Step 1. Set the initial number of scout bee n and probability p_0.

Step 2. Generate two positions randomly named a and b, for each $a < b$.

Step 3. Get a random probability $0 < p < 1$; if $p > p_0$ turn to Step 4; otherwise, turn to Step 5.

```
Procedure: ABC
Input: the initial sequence, the number of scout bee n and p0
Output: the better sequence
Begin
Take n and p0
while i < n
    p = rand[0, 1]
    take a < b
  if p > p0
      Reverse the array between position a and b
  else
      Swap the position of a and b
Output: the better sequence
End
```

Pseudocode 1: Pseudocode of an ABC framework.

Step 4. Reverse the array between positions a and b as a new solution.

Step 5. Swap the position of a and b as a new solution.

Step 6. Calculate the cost of new solution.

Step 7. Keep the best solution to the next iteration and return to Step 2.

The pseudocodes of ABC are shown in Pseudocode 1.

3.3. Global Pheromone Trail Update. The global pheromone updating rule is triggered at the end of iteration to reward tours that are in line with the objective of impedance minimization. This strategy is applied to reinforce the pheromone density on the sets of edges belonging to the inspect tour and to increase the likelihood that this tour will also be selected by other ant agents. The rule of global pheromone updating is given by

$$\tau_{ij} = (1-\rho)\,\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad i, j \in T^{bs}, \tag{20}$$

where

$$\Delta\tau_{ij}^{bs} = \begin{cases} \dfrac{L}{L_{bs}} & i, j \in \text{global best tour} \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

L is a constant initial pheromone. L_{bs} is the cost of the best of all the tours produced by all m agents from the beginning of the iteration. $\rho \in (0, 1]$ is the pheromone evaporation coefficient. To improve the pheromone trail quality, a part of the worst result is removed.

3.4. Local Pheromone Trail Update. In addition to the global pheromone trail updating rule, the selected ants will update the local pheromone trail in the process of passing an arc(i, j). It is opposite to the normal pheromone trail updating rule that increases the pheromone density while ants cross over arcs. The purpose of using the local pheromone trail update rule is to prevent stagnation behavior because the arc becomes less desirable for the following ants. The rule of local pheromone updating is as follows:

$$\tau_{ij} = (1-\xi)\,\tau_{ij} + \xi\tau_0, \tag{22}$$

where τ_0 is a constant at the beginning of pheromone trails and ξ is a user-defined coefficient that lowers the pheromone density of arcs traversed by the intelligent ants.

3.5. Algorithm Flow

Step 1. Get the formulas for solving N_{rv}.

Step 2. Set the initial parameters for the model: set of candidate MCs **R**, set of DPs **S**, set of vehicles **V**, inspecting cost a_r, ordering cost b_r, transportation costs c_r, daily demand d_i, dispatching vehicles cost e_r, fixed (annual) administrative and construction cost f_r, vehicle capacity g, holding cost h, and returning cost k.

Step 3. Parameter setting for HACO is as follows: ants number m, evolution terminate iteration M, pheromone concentration impact factor α, heuristic information pheromones impact factor β, evaporation rate of the pheromone ρ, constant initial pheromone L, and mutation probability array reverse p_0.

Step 4. Using unit matrix τ_{ij}, calculate the probability

$$P_{ij}^{k} = \begin{cases} \dfrac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{k\in\text{allow}(k)}\left[\tau_{ik}\right]^{\alpha}\left[\eta_{ik}\right]^{\beta}} & j \in \text{allow}\,(k) \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

Step 5. Ant solutions generation module: each ant will generate a feasible solution after traversing the DPs.

Step 6. Best ant solution module: after calculating each ant's solution, select the best solution which is known as the iteration best to compare with the global best. Keep the next best solution as the next global best.

```
Procedure: HACO for LIRP
Input: coordinates of nodes, demands and returns of DPs, MC parameters, vehicle capacity, HACO parameters
Output: the best solution (include routes, MCs locations, order times and order size)
Begin
Take max t
while max t < M
    for 1 to m
        Foraging Behavior of Ants
          τ_ij = (1 − ξ)τ_ij + ξτ_0
        Calculate individual total cost Tcost(m)
    end
        Tcost_best = min(Tcost(m))
    for 1 to m/2 Neighbor range
      Scout bee searching the neighbor range
      a = round(rand)
      b = round(rand)
      make a < b
      if p < 0.5
        vech = [vech(1 : a − 1) vech(b : −1 : a) vech((b + 1) : end)];
      else
        exchange(a, b)
      end
        if Tcost(m/2) < Tcost_best
        Tcost_best = Tcost(m/2)
        end
    end
        for 1 to m
            if Tcost(m) > Tcost_ave
              for 1 to n
            τ_ij = (1 − ρ)τ_ij + ρΔτ_ij^bs
              end
            end
        end
end
Output: the best solution
End
```

Pseudocode 2: Pseudocode of the proposed HACO.

Step 7. Scout bee module: random selection probability $0 < p < 1$; if $p > p_0$, turn to random array reverse operation. Otherwise, turn to random swap operation.

Step 8. Pheromone updating module: update the information pheromones as follows:

$$\begin{aligned} \tau_{ij} &= \left(1-\rho\right)\tau_{ij} + \rho\Delta\tau_{ij}^{bs} \quad i, j \in T^{bs}, \\ \tau_{ij} &= \left(1-\xi\right)\tau_{ij} + \xi\tau_0. \end{aligned} \tag{24}$$

Step 9. Termination module: if the parent optimal solution and offspring optimal solution are equal during continuous M generations, stop the algorithm. Otherwise, return to Step 3 after M increments.

Step 10. Output.

The pseudocodes of HACO are shown in Pseudocode 2.

4. Computational Experiments and Results Analysis

In this section, numerical simulations are given to illustrate the performance of HACO compared with the traditional ACO. Both algorithms in this paper are compiled by *Matlab R2014a* and run on a computer with 8 GB main memory and 3.6 GHZ CPU. All instances come from the LRP database in University of Aveiro [36].

4.1. Parameters Discussion. Parameter values selection is crucial to the efficiency of algorithms. An example named Gaskell 67-22 × 5 from the database, which contains the nodes coordinate and the DPs demand, is used to determine the optimal parameter. Gaskell 67 is the instance's name and 22 × 5 means 5 candidate MCs for 22 DPs. The inventory holding cost $h = 2$, the vehicle capacity $g = 500$, transportation costs $c_r = 2$, returning cost $k = 2$, working days $\lambda = 300$, and the delivering cost per unit distance $l = 0.7$. The other parameters of the instance are as follows: $a_r \sim U(16, 20)$;

Table 1: Results with different m (cost: million CNY; iterations: times).

m		Cost			Iterations		
		Mean	Std. dev.	C.V.	Mean	Std. dev.	C.V.
30	ACO	32.4	0.96	0.0295	168.6	30.64	0.1817
	HACO	33.46	1.12	0.0336	125.72	29.02	0.2308
40	ACO	32.13	0.96	0.03	173.48	32.95	0.1899
	HACO	33.37	0.93	0.0277	117.82	15.4	0.1307
50	ACO	32.22	0.74	0.0229	157.19	40.58	0.2581
	HACO	32.96	0.96	0.0292	119.26	17.47	0.1465
60	ACO	32.21	0.67	0.0208	158.12	39.39	0.2491
	HACO	32.72	0.86	0.0263	121.86	17.37	0.1425
70	ACO	31.96	0.71	0.0223	162.88	32.48	0.1994
	HACO	32.77	1.13	0.0344	121.06	23.7	0.1958

$\alpha = 1$, $\beta = 5$, $\rho = 0.1$, $L = 150$, and $p_0 = 0.5$.

Table 2: Results with different α (cost: million CNY; iterations: times).

α		Cost			Iterations		
		Mean	Std. dev.	C.V.	Mean	Std. dev.	C.V.
0.25	ACO	31.92	0.54	0.0168	189.3	73.28	0.3871
	HACO	31.63	0.79	0.025	189.38	69.64	0.3677
0.5	ACO	31.85	0.78	0.0246	181.86	57.34	0.3153
	HACO	31.52	0.91	0.029	182.24	63.94	0.3509
1	ACO	32.22	0.74	0.0229	157.19	40.58	0.2581
	HACO	33.06	1.04	0.0315	123.28	14.02	0.1137
1.25	ACO	32.13	0.79	0.0247	154.24	30.12	0.1953
	HACO	33.11	1.09	0.0331	115.78	11.61	0.1002
1.5	ACO	32.14	0.9	0.0279	154.96	25.68	0.1658
	HACO	32.97	1.11	0.0335	116.2	10.23	0.0881

$m = 50$, $\beta = 5$, $\rho = 0.1$, $L = 150$, and $p_0 = 0.5$.

Table 3: Results with different β (cost: million CNY; iterations: times).

β		Cost			Iterations		
		Mean	Std. dev.	C.V.	Mean	Std. dev.	C.V.
3	ACO	31.91	0.91	0.0286	192.78	42.08	0.2183
	HACO	33.06	1.19	0.0359	148.8	54.84	0.3686
4	ACO	32.19	0.8	0.0249	170.2	41.89	0.2461
	HACO	33.05	1.14	0.0344	126	42.17	0.3347
5	ACO	32.22	0.74	0.0229	157.19	40.58	0.2581
	HACO	32.96	0.96	0.0292	119.26	17.47	0.1465
6	ACO	32.1	0.77	0.0241	159.38	31.99	0.2007
	HACO	33.12	1.19	0.0358	119.86	22.4	0.1869
7	ACO	32.42	0.84	0.0258	154.69	23.92	0.1546
	HACO	33.25	1.07	0.0322	111.68	8.62	0.0772

$\alpha = 1$, $m = 50$, $\rho = 0.1$, $L = 150$, and $p_0 = 0.5$.

$b_r \sim U(16, 20)$; $c_r \sim U(6, 10)$; $e_r \sim U(21, 25)$; $q_i \sim U(12, 25)$; and $w_i \sim U(2, 5)$.

The parameters of algorithm are initialized as follows: ant's number $m = 50$, evolution terminate iteration $M = 100$, pheromone concentration impact factor $\alpha = 1$, heuristic information pheromones impact factor $\beta = 5$, evaporation rate of the pheromone $\rho = 0.1$, constant $L = 150$, mutation probability array reverse $p_0 = 0.5$.

We run the program 50 times on the same computer. The performance of ACO and HACO varies with the different values of the parameters, which are shown in Tables 1–6. In these tables, the symbol C.V. means the coefficient of variation.

Tables 1–6 represent the parameters' effect on the objective function values. The data was normalized through two dimensions, that is, cost and iterations, and three indicators,

Table 4: Results with different ρ (cost: million CNY; iterations: number).

ρ		Cost			Iterations		
		Mean	Std. dev.	C.V.	Mean	Std. dev.	C.V.
0.1	ACO	32.22	0.74	0.0229	157.19	40.58	0.2581
	HACO	32.12	0.93	0.0289	138.4	21.02	0.1518
0.2	ACO	32.38	0.87	0.0269	133.27	20.57	0.1543
	HACO	32.7	0.97	0.0296	124.12	13.01	0.1048
0.3	ACO	32.65	0.94	0.0289	132.8	31.09	0.2341
	HACO	32.75	1.45	0.0443	122.1	13.09	0.1072
0.4	ACO	33.03	0.94	0.0284	131.86	28.76	0.2181
	HACO	33.31	1.1	0.033	122.98	20.2	0.1643
0.5	ACO	33.41	0.87	0.0261	157.27	67.72	0.4306
	HACO	33.55	1.08	0.0323	116.24	15.44	0.1328

$\alpha = 1$, $m = 50$, $\beta = 5$, $L = 150$, and $p_0 = 0.5$.

Table 5: Results with different L (cost: million CNY; iterations: times).

L		Cost			Iterations		
		Mean	Std. dev.	C.V.	Mean	Std. dev.	C.V.
50	ACO	31.99	0.72	0.0224	159.72	33.39	0.21
	HACO	33.07	1.22	0.0369	123.7	25.96	0.2098
100	ACO	31.9	0.67	0.0209	159.8	30.73	0.19
	HACO	32.88	1.09	0.033	121.63	18.91	0.1555
150	ACO	32.22	0.74	0.0229	157.19	40.58	0.2581
	HACO	33.02	1.05	0.0319	121.14	23.2	0.1915
200	ACO	32.11	0.8	0.0251	167.66	38.61	0.23
	HACO	32.9	1.12	0.0339	120.8	22.05	0.1825
250	ACO	32.08	0.87	0.0271	164.08	34.16	0.21
	HACO	33.1	1.09	0.033	121.87	28.95	0.2376

$\alpha = 1$, $m = 50$, $\beta = 5$, $\rho = 0.1$, and $p_0 = 0.5$.

Table 6: Results with different p_0 (cost: million CNY; iterations: times).

p_0	Cost			Iterations		
	Mean	Std. dev.	C.V.	Mean	Std. dev.	C.V.
0.3	33.07	1.22	0.0369	123.70	25.96	0.2098
0.4	32.88	1.09	0.0330	121.63	18.91	0.1555
0.5	33.02	1.05	0.0319	121.14	23.20	0.1915
0.6	32.90	1.12	0.0339	120.80	22.05	0.1825
0.7	33.10	1.09	0.0330	121.87	28.95	0.2376

$\alpha = 1$, $m = 50$, $\beta = 5$, $\rho = 0.1$, and $L = 150$.

that is, mean, standard deviation, and coefficient of variation. Actually, in order to find the minimal cost, we usually take the parameter values, where the cost is lower and more stable. From the discussion, we found that HACO reaches the best performance when $m = 60$, $\alpha = 0.5$, $\beta = 5$, $\rho = 0.1$, $L = 100$, and $p_0 = 0.4$, while ACO reaches the best performance when $m = 70$, $\alpha = 0.5$, $\beta = 3$, $\rho = 0.1$, and $L = 100$.

4.2. Computational Experiment. To get a reliable conclusion, we run another 50 times on the same computer with the best parameter values in Gaskell 67-22 × 5. One of the best solutions of objective function in the 50 experiments of HACO is 30.2 million CNY. Table 7 shows the solution. MCs were established at MC1, MC2, and MC5 with five vehicles distribution routes. Figure 2 shows topological structure of the closed-loop supply chain.

Figure 3 shows their trend of optimal objective function values along with iterations. The fluctuation curves of optimal objective function value are varied by different algorithm, which are shown in Figures 4(a) and 4(b).

As shown in Figure 3, the cost and iterations of HACO are lower than ACO; and in Figure 4, the range and mean value of

Table 7: The solution of Gaskell 67-22 × 5.

MC	Routing number	Routing	Order times
MC1	V1	1 → 12 → 13 → 9 → 10 → 26 → 1	17
	V2	1 → 16 → 15 → 18 → 14 → 1	18
MC2	V3	2 → 23 → 24 → 27 → 25 → 17 → 2	37
MC5	V4	5 → 22 → 20 → 21 → 8 → 19 → 5	24
	V5	5 → 7 → 6 → 11 → 5	18

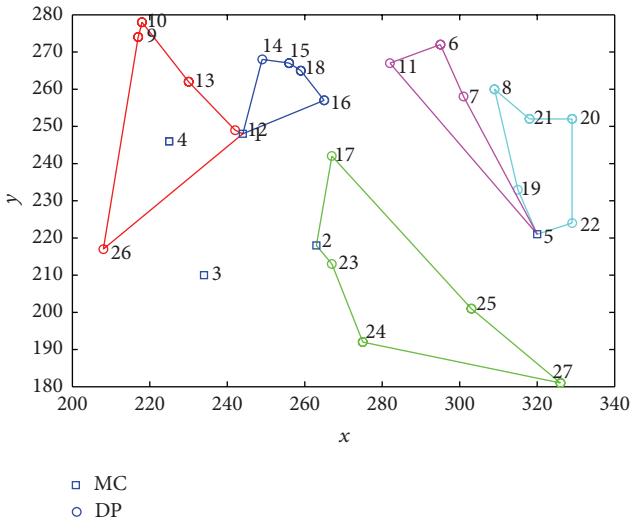


Figure 2: Topological structure of the network.

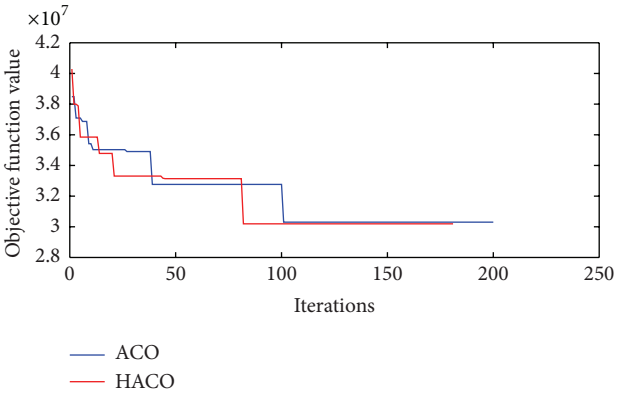
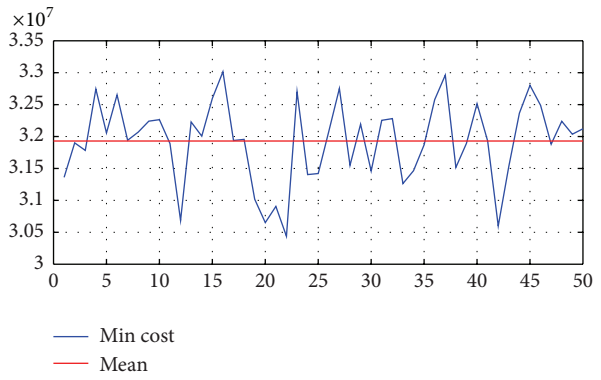
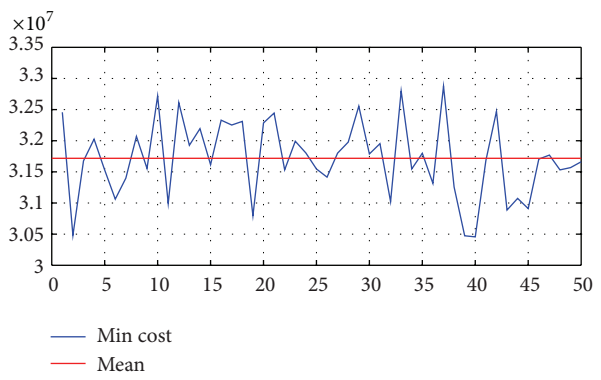


Figure 3: Trends of objective function value.



(a) ACO



(b) HACO

Figure 4: The fluctuation curve of optimal objective function value.

the minimum cost of HACO are also lower than ACO, which both imply that HACO is more efficient than ACO in solving the LIRP.

4.3. Extended Experiments. In this section, a series of instances are given to show that HACO is more efficient and stable than classical software and ACO. In order to ensure the demands of DPs are not more than the vehicle capacity, we need to enumerate some instances. In this paper, the daily demands are set as 1/10 of corresponding demands of the database.

As we know, Lingo is a representative classical optimization software tool. Thus we used *Lingo 11.0* to solve the problem by using a small-sized instance named Peal 183-12 × 2 and two medium-sized instance named Gaskell 67-22 × 5 and Gaskell 67-36 × 5; the results are shown in Table 8.

Each instance was run 50 times by HACO and ACO with their optimized parameters values, respectively; the results are shown in Tables 9 and 10.

4.4. Result Analysis. According to Table 8, we found that (1) for the small-sized instance, HACO can obtain better result than Lingo within less time and (2) for medium-sized instances, Lingo cannot get the global optimization within 1 hour, while HACO can solve the problem in a short time.

Observe, from Tables 9 and 10, that HACO is more efficient than ACO for the following reasons. (1) The cost of HACO is significantly lower than ACO ($p < 0.05$); (2) the difference of the number of iteration between HACO and ACO is not significant ($p > 0.05$); (3) HACO is more stable than ACO as the coefficient of variation (C.V.) is lower. To sum up, our algorithm reduces the cost with the same number of iterations compared with ACO.

By improving pheromone updates and bee colony searching, we improve the solution quality of the algorithm and make it useful as a guide for the ant searching process. Observed from the results of numerical simulations, HACO can get better result with a fewer number of iterations. Hence,

Table 8: Comparisons between HACO and Lingo.

Instance name	Perl 183-12 × 2		Gaskell 67-22 × 5		Gaskell 67-36 × 5	
	CPU time	Cost	CPU time	Cost	CPU time	Cost
Lingo	523 s	715392	>1 hour	\	>1 hour	\
HACO	24.5 s	709152	44 s	32691202	99 s	31550446

Table 9: Optimal objective function values of two algorithms (CNY).

Instance name	Algorithm	Mean	Std. dev.	C.V.	t value	Significance test p value (sig. 1-tailed)
Perl 183-12 × 2	ACO	711347.1	6979.11	0.0098	1.6838	0.048
	HACO	709237.9	5454.48	0.0077		
Gaskell 67-22 × 5	ACO	31929818.3	627210.66	0.0196	1.7118	0.045
	HACO	31718084.1	609606.04	0.0192		
Gaskell 67-36 × 5	ACO	3288921.3	43585.25	0.0133	1.6602	0.050
	HACO	3275052	39868.81	0.0122		
Perl 183-55 × 15	ACO	285145.3	3779.15	0.0133	1.8391	0.034
	HACO	283831.6	3351.00	0.0118		
Christofides 69-75 × 10	ACO	419695.8	4948.15	0.0118	2.5020	0.007
	HACO	417298.6	4627.32	0.0111		
Perl 183-85 × 7	ACO	486011.0	4503.39	0.0093	1.8080	0.037
	HACO	484424.1	4270.44	0.0088		
Christofides 69-100 × 10	ACO	458235.6	6387.00	0.0139	1.6796	0.048
	HACO	456441.6	5317.70	0.0117		

Table 10: Iterations of two algorithms (times).

Instance name	Algorithm	Mean	Std. dev.	C.V.	t value	Significance test p value (sig. 1-tailed)
Perl 183-12 × 2	ACO	193.32	78.70	0.4071	0.6323	0.264
	HACO	183.90	70.04	0.3808		
Gaskell 67-22 × 5	ACO	191.34	86.36	0.4513	0.7831	0.218
	HACO	180.70	43.32	0.2397		
Gaskell 67-36 × 5	ACO	254.44	76.84	0.3020	0.2012	0.421
	HACO	251.84	49.42	0.1962		
Perl 183-55 × 15	ACO	241.74	77.29	0.3197	0.1000	0.460
	HACO	240.46	47.33	0.1968		
Christofides 69-75 × 10	ACO	245.24	79.10	0.3226	1.2340	0.110
	HACO	229.92	62.30	0.2710		
Perl 183-85 × 7	ACO	247.18	94.89	0.3839	0.4355	0.332
	HACO	240.80	65.75	0.2730		
Christofides 69-100 × 10	ACO	248.06	97.18	0.3918	0.2627	0.397
	HACO	244.08	65.48	0.2681		

comparing with ACO, HACO is adopted as a better approach in solving this LIRP with MQDR.

5. Conclusion and Future Research

With the development of e-commerce, customers' return keeps a high rate with MQDR, which can be reentered into markets after being repackaged or recovered. In this research, we built a closed-loop LIRP model considering both quality defect returns and nondefect returns; we call it MQDR in this paper. We perform an extensive computational study and observe the following interesting results.

(1) Considering MQDR are computationally beneficial for the formulation presented, the MQDR and closed-loop pattern with returns are features of the proposed

problem in e-commerce, which is never considered in previous work.

(2) Since the evolutionary computation algorithm has been proved successfully in tackling NP-hard problem, a hybrid algorithm is proposed by combining ACO algorithm and ABC algorithm to solve the LIRP. HACO integrated the scout bee searching phase into the ACO to improve the global searching ability.

(3) The performance of HACO is evaluated by using the instances in the LRP database, and HACO outperforms ACO on convergence, optimal solution, and computing stability. This numerical study shows the efficiency and effectiveness of the solution method.

However, developing other elements for the LIRP will lead to further research directions. And analyzing the model under the dynamic demand of customs and a time-varying demand can be a valuable subject. The design of experiments and verification by discrete dynamics simulation should be established. Fruit fly optimization algorithm (FOA) as one of the best EC algorithms has attracted the attention of various researchers [37]. It is important to apply these models and algorithms to the operation and management of enterprises to improve the decision-making efficiency of e-commerce logistics system.

Competing 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. 71171093,71471073, and 71571082) and self-determined research funds of CCNU from the colleges' basic research and operation of MOE (nos. CCNU14Z02016 and CCNU15A02046).

References

[1] "Ecommerce sales worldwide reach $1.471 trillion in 2014," http://www.emarketer.com/Article/Worldwide-Ecommerce-Sales-Increase-Nearly-20-2014/1011039.

[2] J. Mostard and R. Teunter, "The newsboy problem with resalable returns: a single period model and case study," *European Journal of Operational Research*, vol. 169, no. 1, pp. 81–96, 2006.

[3] F. Zeng, Y. Chi, and J. Zheng, "The newsvendor problem with different delivery time, resalable returns, and an additional order," *Mathematical Problems in Engineering*, vol. 2015, Article ID 505268, 10 pages, 2015.

[4] R. H. Ballou, *Business Logistics Management*, Prentice Hall, New York, NY, USA, 1992.

[5] J. Perl and S. Sirisoponsilp, "Distribution networks: facility location, transportation and inventory," *International Journal of Physical Distribution & Materials Management*, vol. 18, no. 6, pp. 18–26, 1988.

[6] R. H. Ballou and J. M. Masters, "Commercial software for locating warehoused and other facilities," *Journal of Business Logistics*, vol. 14, no. 2, pp. 70–107, 1993.

[7] S. J. Erlebacher and R. D. Meller, "The interaction of location and inventory in designing distribution systems," *IIE Transactions*, vol. 32, no. 2, pp. 155–166, 2000.

[8] M. S. Daskin, C. R. Coullard, and Z. J. Shen, "An inventory-location model: formulation, solution algorithm and computational results," *Annals of Operations Research*, vol. 110, no. 1–4, pp. 83–106, 2002.

[9] 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.

[10] S.-H. Liao, C.-L. Hsieh, and P.-J. Lai, "An evolutionary approach for multi-objective optimization of the integrated location-inventory distribution network problem in vendor-managed inventory," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6768–6776, 2011.

[11] A. Balakrishnan, J. E. Ward, and R. T. Wong, "Integrated facility location and vehicle routing models: recent work and future prospects," *American Journal of Mathematical and Management Sciences*, vol. 7, no. 1-2, pp. 35–61, 1987.

[12] H. Min, V. Jayaraman, and R. Srivastava, "Combined location-routing problems: a synthesis and future research directions," *European Journal of Operational Research*, vol. 108, no. 1, pp. 1–15, 1998.

[13] G. Nagy and S. Salhi, "Location-routing: issues, models and methods," *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.

[14] L. M. A. Chan, A. Federgruen, and D. Simchi-Levi, "Probabilistic analyses and practical algorithms for inventory-routing models," *Operations Research*, vol. 46, no. 1, pp. 96–106, 1998.

[15] A. J. Kleywegt, V. S. Nori, and M. W. P. Savelsbergh, "The stochastic inventory routing problem with direct deliveries," *Transportation Science*, vol. 36, no. 1, pp. 94–118, 2002.

[16] D. Adelman, "A price-directed approach to stochastic inventory/routing," *Operations Research*, vol. 52, no. 4, pp. 499–514, 2004.

[17] 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.

[18] 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, pp. 941–950, 2003.

[19] 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.

[20] Z.-J. M. 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.

[21] A. A. Javid and N. Azad, "Incorporating location, routing and inventory decisions in supply chain network design," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 5, pp. 582–597, 2010.

[22] A. Ahmadi-Javid and A. H. Seddighi, "A location-routing-inventory model for designing multisource distribution networks," *Engineering Optimization*, vol. 44, no. 6, pp. 637–656, 2012.

[23] W. J. Guerrero, C. Prodhon, N. Velasco, and C. A. Amaya, "Hybrid heuristic for the inventory location-routing problem with deterministic demand," *International Journal of Production Economics*, vol. 146, no. 1, pp. 359–370, 2013.

[24] Y. Zhang, M. Y. Qi, L. X. Miao, and E. Liu, "Hybrid metaheuristic solutions to inventory location routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, no. 1, pp. 305–323, 2014.

[25] N. Nekooghadirli, R. Tavakkoli-Moghaddam, V. R. Ghezavati, and S. Javanmard, "Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics," *Computers and Industrial Engineering*, vol. 76, no. 1, pp. 204–221, 2014.

[26] W. J. Guerrero, C. Prodhon, N. Velasco, and C. A. Amaya, "A relax-and-price heuristic for the inventory-location-routing problem," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 129–148, 2015.

[27] Y. H. Li, H. Guo, L. Wang, and J. Fu, "A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under E-supply chain environment," *The Scientific World Journal*, vol. 2013, Article ID 125893, 10 pages, 2013.

[28] B. L. Liu, H. Chen, Y. H. Li, and X. Liu, "A pseudo-parallel genetic algorithm integrating simulated annealing for stochastic location-inventory-routing problem with consideration of returns in e-commerce," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 586581, 15 pages, 2015.

[29] J. A. Petersen and V. Kumar, "Perceived risk, product returns, and optimal resource allocation: evidence from a field experiment," *Journal of Marketing Research*, vol. 52, no. 2, pp. 268–285, 2015.

[30] Y. Niu, S. Wang, J. He, and J. Xiao, "A novel membrane algorithm for capacitated vehicle routing problem," *Soft Computing*, vol. 19, no. 2, pp. 471–482, 2014.

[31] M. Ball, L. Bodin, and R. Dial, "A matching based heuristic for scheduling mass transit crews and vehicles," *Transportation Science*, vol. 17, no. 1, pp. 4–31, 1983.

[32] M. Dror, G. Laporte, and P. Trudeau, "Vehicle routing with stochastic demands: properties and solution frameworks," *Transportation Science*, vol. 23, no. 3, pp. 166–176, 1989.

[33] Y. Dumas, J. Desrosiers, and F. Soumis, "The pickup and delivery problem with time windows," *European Journal of Operational Research*, vol. 54, no. 1, pp. 7–22, 1991.

[34] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[35] K. Dervis and B. Bahriye, "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.

[36] "Location-Routing Problems (LRP)," http://sweet.ua.pt/~iscf143/_private/SergioBarreto HomePage.htm.

[37] L. Wang, Y. Shi, and S. Liu, "An improved fruit fly optimization algorithm and its application to joint replenishment problems," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4310–4323, 2015.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2016, Article ID 5348203, 9 pages
http://dx.doi.org/10.1155/2016/5348203

Research Article

A Genetic Algorithm with Location Intelligence Method for Energy Optimization in 5G Wireless Networks

Ruchi Sachan, Tae Jong Choi, and Chang Wook Ahn

Department of Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Republic of Korea

Correspondence should be addressed to Chang Wook Ahn; cwan@skku.edu

Received 11 December 2015; Revised 18 April 2016; Accepted 11 May 2016

Academic Editor: Ahmed Kattan

The exponential growth in data traffic due to the modernization of smart devices has resulted in the need for a high-capacity wireless network in the future. To successfully deploy 5G network, it must be capable of handling the growth in the data traffic. The increasing amount of traffic volume puts excessive stress on the important factors of the resource allocation methods such as scalability and throughput. In this paper, we define a network planning as an optimization problem with the decision variables such as transmission power and transmitter (BS) location in 5G networks. The decision variables lent themselves to interesting implementation using several heuristic approaches, such as differential evolution (DE) algorithm and Real-coded Genetic Algorithm (RGA). The key contribution of this paper is that we modified RGA-based method to find the optimal configuration of BSs not only by just offering an optimal coverage of underutilized BSs but also by optimizing the amounts of power consumption. A comparison is also carried out to evaluate the performance of the conventional approach of DE and standard RGA with our modified RGA approach. The experimental results showed that our modified RGA can find the optimal configuration of 5G/LTE network planning problems, which is better performed than DE and standard RGA.

1. Introduction

The green domain is a new stage which aims to protect Earth and contribute to reducing the global warming by efficiently optimizing the energy consumption. Thus, the need for energy efficient wireless networks has drawn significant attention and focuses on the need to cut operating expenses and power usage of the telecommunications infrastructure, where radio networks represent about 80% of energy consumption. Furthermore, it is widely known that base stations (BSs) consume a significant amount of the energy (above 50%) in a cellular network [1, 2], which requires optimization of the transmission power and location of a BS regarding green aspects as shown in Figure 1.

The current 3G and 4G communication technologies were introduced to fulfill the massive demand for enhancing the speed of data traffic. Although the current communications technology has progressed impressively, it is still facing the increasing demands due to the development of smart devices. For this reason, various intensive studies towards 5G networks are being developed beyond the current 4G/IMT-Advanced standards and are moving towards the next phase of mobile communication. The most important requirement for the development of 5G network is the enhanced data traffic; that is, it has to support robustly an exponentially increasing number of devices [3]. Moreover, Long-Term Evolution (LTE) which is expected to be used with 5G networks has to deal with the reduced cell size of a BS [4], which leads to an increase in the number of BSs and raises a concern about increasing energy consumption of BSs.

Fortunately, 5G networks would benefit from the position information and fittingly guide the wireless network designs and optimization. There are many ways to find precise location information in wireless networks along with related distances, velocities, angles, delays, and predictable user behavior [5] in 5G networks. The information obtained from location-aware technology can be used to address numerous issues by implementing sharing and coexistence approaches to the challenges in 5G networks based on the user's position. By getting more accurate information of the users, power

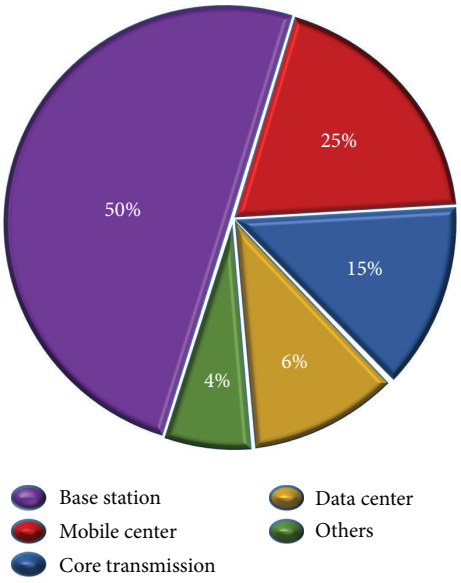


Figure 1: The energy consumption ratio in cellular networks.

efficiency can be improved by placing the sufficient base stations in a reasonable position based on the user's behavior. Overload and delay can also be reduced by using location-aware information.

In this paper, Real-coded Genetic Algorithm (RGA) was modified to allocate the base stations efficiently in a dense urban area regarding green aspects based on the user's optimal position. In other words, we modified RGA to solve our BSs allocation problem efficiently. We found out that standard RGA cannot converge to an optimal solution because its offspring is created by shuffling all chromosomes of its parents. By introducing BS Crossover Rate (BCR) to RGA and slight modification of mutation, our modified RGA can perform better than not only standard RGA but also differential evolution (DE) algorithm. Thus, the key contribution of this paper is that we modified RGA-based method to find the optimal configuration of BSs not only by just offering an optimal coverage of underutilized BSs but also by optimizing the amounts of power consumption.

In this study, we introduced a literature survey in Section 2 which dealt with optimizing the energy consumption in cellular networks. A system model which provided a description of the general framework for network planning in the context of 5G networks was explained in Section 3. The idea of implementing a modified RGA for optimizing the energy consumption of BSs was presented in Section 4. The experimental results obtained from our proposed method with the conclusion to this paper were illustrated in Sections 5 and 6, respectively.

2. Related Work

To date, many researchers have rigorously studied network design problems such as planning and optimizing performance in cellular networks with a considerable amount of published work, specifically on Universal Mobile Telecommunications System (UMTS) networks, LTE networks, and 5G networks. These studies aimed to estimate the optimal configuration for the locations of BSs, power requirement of each transmitter, antenna heights, relaying, and load balancing [6–8]. For instance, a variation of simulated annealing algorithm was proposed to find the optimal design for least cost and full coverage planning in Single Frequency Networks (SFNs) [9]. In this paper, it was stated that the worst solution found by the stochastic optimization algorithm based on numerical formulation is more efficient than the best solution found by the conventional coverage planning algorithm.

In [10], the authors stated that avoiding the maximum throughput regarding the minimizing energy consumption for designing an initial cell is efficient. When a coverage area is to be planned, network designers should estimate the calculations for neighbor interference, which is known as in-band Inter-Cell Interference (ICI). These methods are useful to avoid ICI while minimizing cell edge throughput.

In the network planning problem for WiMAX, the authors of [11] have determined a location-awareness issue by using the Evolutionary Algorithms (EAs) such as Artificial Bee Colony (ABC) and genetic algorithm (GA) to meet the traffic and coverage requirements for the targeted base stations efficiently. For minimizing the interference among cells and reducing energy consumption, the authors have divided a genetic algorithm into APS-GA (Genetic Algorithm with Adaptive Population Size) and FPS-GA (Genetic Algorithm with Fixed Population Size) to resolve the same problem for comparing these two algorithms efficiently. After the comparison of APS-GA and FPS-GA with ABC algorithms, it was observed that the ABC algorithm required minimum computational efforts, that is, less population and fewer evaluations, compared to the APS-GA and FPS-GA, while the balanced load could not efficiently satisfy the connected users.

In [12], the authors proposed an algorithm for joint uplink and downlink radio planning in a UMTS with the objective of minimizing the total energy consumption. The authors successively executed and subdivided the problem into two segments. Firstly, the authors tried to find the optimal positions of a fixed number of UMTS BSs in the given area of interest for targeting the optimization problem for best locations of BSs. The study aimed to minimize the total downlink power expense and at the same time the uplink outage that depends on the power abilities of Mobile Stations (MSs) under different restraints which sustain an acceptable Quality of Service (QoS) and satisfy the energy budget. In a second phase, they proposed an algorithm to select the minimal set of BSs with fixed locations based on the site awareness prospects.

In the perspective of energy consumption, some recent researchers [13, 14] have also discussed a method of achieving optimum power savings by switching off traffic underloaded BSs (eNBs) in LTE technologies. In the article [13], the vast amount of energy conservation is estimated by disabling unnecessary cells due to low traffic. As outlined by different studies that mostly tackle the sleep mode at mobile user's side, there is trade-off between outage of users and energy saving [14].

In [15], the authors have proposed a scheme on resource allocation for next-generation 5G networks. They calculated Signal to Interference Noise Ratio (SINR) based on the estimated path loss for each BSs and MSs pair. The power received from each user was compared with the given threshold value for the coverage area of macrocell and microcell as these cells come in the new features of the heterogeneous cell for the modern communication of LTE and LTE-Advance or beyond. In the conclusion, the authors had just proposed a scheme based on the radio planning for giving a good coverage to the users, but they did not care about optimizing the power consumption. Our paper has considered not only a good coverage but also power consumption by using the application of the EAs. There is scope for improving and optimizing network planning concerning the green aspects by using the new features available in 5G networks.

3. System Model

The system model assumes that we consider occupying the area of $[W \times H]$ Km^2 for 5G/LTE networks, where BSs can be installed in that given area with a set of candidate sites $H = \{h_1, h_2, h_3, \ldots, h_M\}$, which is given in this scenario. In order to place the BSs we need an installing cost which is associated with each of the candidate sites $C = \{c_1, c_2, c_3, \ldots, c_M\}$. The number of BSs in our simulation is denoted by K and a set of BSs is denoted by $B = \{b_1, b_2, b_3, \ldots, b_K\}$.

As for the power of a transmitter, it is given that a BS has the transmission power in the range of 0.1 to 10 Watt. In reality, the value of antenna gain depends on the manufacture but, in our paper, it is assumed to be 18 dBi, and frequency is used as 1800 MHz [3, 16]. We employ the Cost-231 HATA urban propagation model as this model is also known as a radio propagation model, but it extends the urban HATA model to cover a more enlarged range of frequencies [17, 18]. The coverage probability in the area around the location h_i with a threshold is less than the SINR. The SINR is calculated by using (1), where M_g is the MHA gain, P_t is the transmission power, and I and N refer to the interference and noise, respectively. Consider

$$\text{SINR} = \frac{M_g \cdot P_t}{I + N}. \tag{1}$$

After calculating the value of SINR, the path loss (PL) is determined by

$$\text{PL [dB]} = P_t + G_t - L_b - \text{SINR}, \tag{2}$$

where G_t is the transmitter's antenna gain and L_b is the body loss in dB. Also, the coverage area of a BS is formulated by

$$\text{CA}_{b_i} = \sqrt[3]{3}\left(\frac{R^2}{2}\right), \tag{3}$$

where R is the cell radius. The coverage probability in that area around the location h_i with threshold T is

$$P_c\left(h_i\right) = P\left(\text{SINR}\left(h_i\right) > T\right). \tag{4}$$

Table 1: Decision variables.

Q_i	Available transmit power of a base station b_i [0.1 to 10.0] Watt
X_i	Location of a base station b_i in x-axis
Y_i	Location of a base station b_i in y-axis

4. The Proposed Algorithm

The application of EAs such as GA, DE, and RGA is a stochastic exploration technique for solving both constrained and unconstrained optimization problems, which are based on the natural selection. This procedure drives the biological evolution, in such a manner that a population of individual solutions is deceptively modified. At each phase, EA probabilistically selects promising individuals from the current population to be parents and then uses them to produce their offspring for the next-generation employing crossover and mutation mechanisms. As such, the application of EA provides satisfactory solutions to NP-hard optimization problems. Additionally, EA is also used to solve many practical problems such as finding an optimal position for a BS [9, 19] in a given area of interest.

Although EAs can solve NP-hard and several practical problems efficiently, designing the structure of EAs properly to a problem is required to achieve optimal performance. The problem that we tried to solve is as follows. Users are located in several dense areas, called urban areas, and we try to give service to as many users as possible by using least number of BSs and least transmit power by locating BSs to optimal positions. To solve this problem, we applied RGA because all the decision variables (transmit power and location) are continuous values. However, standard RGA could not find an optimal solution because the crossover of standard RGA shuffles all of the chromosomes, which makes a huge difference between parents and offspring. Besides, the mutation operator of standard RGA uses high standard deviation values, which also makes a huge difference. To solve this problem to make RGA perform better to solve network planning problem, we modified RGA by introducing BS Crossover Rate (BCR) to shuffle less and using small standard deviation values. These modifications make RGA outperform not only standard RGA but also DE approach in simulation results.

4.1. Encoding. The design of chromosomes is the essential aspect of EAs. The chromosome of EA is a set of decision variables that represents candidate solutions to an objective problem. As usual, a set of chromosomes is a possible solution to the target problem. We have to consider the design of chromosomes carefully for gaining a better representation of a solution to an objective problem while applying stochastic methods. The population consists of a set of individuals that contains the structure of chromosomes.

In this paper, the decision variables of our problem are given in the available transmit power (Q_i) and location of a BS (X_i, Y_i). In our chromosome representation, we used only continuous values. The proposed algorithm uses these decision variables for chromosomes as described in Table 1.

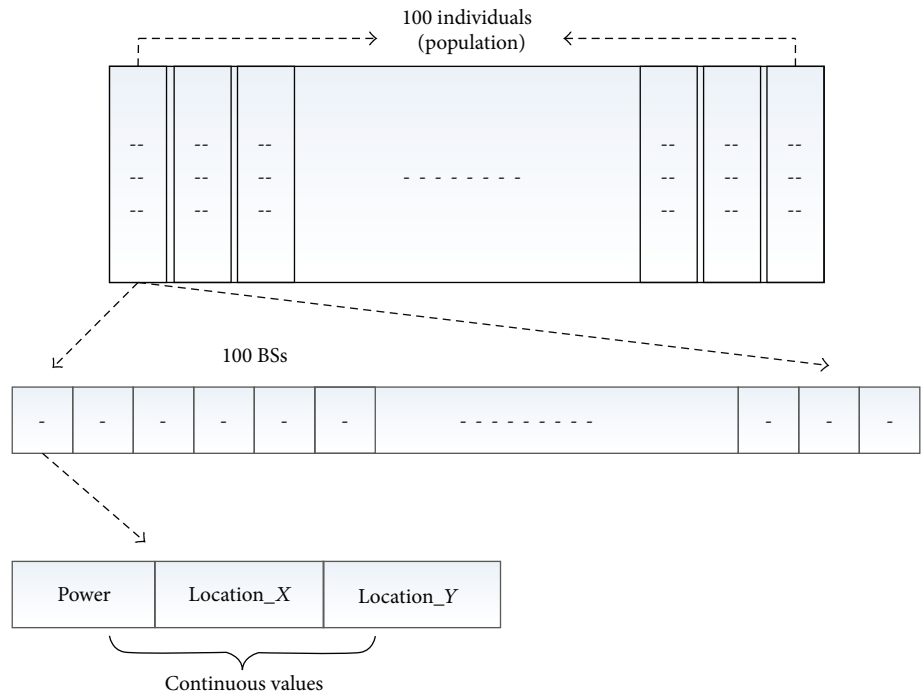


Figure 2: The structure of chromosomes.

In our encoding, the population consists of P individuals and each individual is composed by K BSs and one BS has three decision variables. In other words, a set of K BSs falls within one individual, and a set of P individuals falls within one population. Figure 2 shows how the structure of a population and individuals is organized in our encoding. The values that are out of the ranges given in this scenario for the BSs will be penalized by the given condition in an evaluation operator. By applying this mechanism, the modified RGA can find an optimal configuration.

4.2. Genetic Operations. In general, the operators of EAs consist of selection, crossover, and mutation. We apply roulette wheel selection for selecting potentially valuable solutions for recombination. According to the number of times, the roulette wheel is twisted equally to base on the size of the population and divides from the natural way. Each time the wheel stops it gives the fitter individual the greatest chance of being selected for the next generation and succeeding the mating pool. It can produce faster convergence speed, which has more potential to find fast optimal solutions.

The basic idea behind crossover is to increase genetic diversity. In this mechanism, it exchanges parent individuals' chromosomes and produces their offspring to make better individuals than both of their parents if they take the best characteristics from each of the parents. In our work, we applied box crossover. The upper bound and lower bound of two-dimensional bounded areas evaluate the range of the possible offspring. Box crossover is also a proper feature of RGA to maintain their population diversity efficiently. What we have modified in the standard RGA is that we introduce BCR by applying modified crossover in our proposed modified RGA means MRGA. As mentioned earlier, we found out that standard RGA cannot converge to an optimal solution because its offspring is created by shuffling all chromosomes of its parents. In our modified RGA, we are not shuffling every chromosome of an individual to the next generation by using BCR. Therefore, BCR prevents shuffle of all of the chromosomes, which makes a small difference between parents and offspring. The modified crossover is performed as Algorithm 1. In the Algorithm, the tilde indicates offspring.

Mutation is the part of a genetic operator which is also used to increase genetic diversity from one generation of a population to the next generation. It is meant to break frequently few chromosomes of a population to overcome a local optimum. We have applied slightly modified mutation in our modified RGA. The modified mutation is performed as Algorithm 2.

D value contains a difference between standard RGA's mutation and our modified RGA. In standard RGA, D value is fixed to use 10, while the modified RGA uses 100. Similar to BCR, this modification makes a small difference between parents and offspring, not a huge difference.

After crossover and mutation, our modified RGA conducts replacement operator. In replacement operator, if offspring has better fitness than a randomly selected individual from the population, then offspring replaces the individual. Otherwise, offspring discarded. The overall procedures of the proposed RGA are described in Algorithm 3.

4.3. Fitness Evaluation. A fitness function is used to help get a solution from the evaluation of chromosomes for the survival

(1) **Begin**
(2) r_1 is a random individual from selected individuals
(3) r_2 is a random individual from selected individuals
(4) $r_{1,m}$ is a selected base station based on roulette wheel from r_1
(5) $r_{2,n}$ is a selected base station based on roulette wheel from r_2
(6) **For** $i = 0$ to $i = P$
(7) **For** $j = 0$ to $i = K$
(8) **If** a random number from [0, 1] is less than BSR **Then**
(9) $\widetilde{Q_{i,j}}$ = a random real number in $[\min(Q_{r_{1,m}}, Q_{r_{2,n}}), \max(Q_{r_{1,m}}, Q_{r_{2,n}})]$
(10) $\widetilde{X_{i,j}}$ = a random real number in $[\min(X_{r_{1,m}}, X_{r_{2,n}}), \max(X_{r_{1,m}}, X_{r_{2,n}})]$
(11) $\widetilde{Y_{i,j}}$ = a random real number in $[\min(Y_{r_{1,m}}, Y_{r_{2,n}}), \max(Y_{r_{1,m}}, Y_{r_{2,n}})]$
(12) **Else**
(13) $\widetilde{Q_{i,j}} = Q_{r_{1,m}}$
(14) $\widetilde{X_{i,j}} = X_{r_1,m}$
(15) $\widetilde{Y_{i,j}} = Y_{r_1,m}$
(16) **End If**
(17) **End For**
(18) **End For**
(19) **End**

Algorithm 1: The modified crossover.

(1) **Begin**
(2) **For** $i = 0$ to $i = P$
(3) **For** $j = 0$ to $i = K$
(4) **If** a random number from [0, 1] is less than M_r **Then**
(5) $\widetilde{Q_{\mathrm{i,j}}} = \widetilde{Q_{\mathrm{i,j}}} + N(0, (Q_{\max} - Q_{\min})/D)$
(6) **End If**
(7) **If** a random number from [0, 1] is less than M_r **Then**
(8) $\widetilde{X_{\mathrm{i,j}}} = \widetilde{X_{\mathrm{i,j}}} + N(0, (X_{\max} - X_{\min})/D)$
(9) **End If**
(10) **If** a random number from [0, 1] is less than M_r **Then**
(11) $\widetilde{Y_{\mathrm{i,j}}} = \widetilde{Y_{\mathrm{i,j}}} + N(0, (Y_{\max} - Y_{\min})/D)$
(12) **End If**
(13) **End For**
(14) **End For**
(15) **End**

Algorithm 2: The modified mutation.

(1) **Begin**
(2) Initialize Users
(3) Initialize Population
(4) Evaluate Population
(5) **While** Termination criteria does not met **Then**
(6) SelectedIndividuals = RouletteWheel(Population)
(7) Offspring = ModifiedBoxCrossover(SelectedIndividuals) (Algorithm 1)
(8) Offspring = ModifiedMutation(Offspring) (Algorithm 2)
(9) Replacement(Population, Offspring)
(10) **End While**
(11) **End**

Algorithm 3: Pseudocode of modified RGA.

Table 2: Simulation parameters.

Carrier frequency	15 GHz
Frame structure	FDD
Population size	100
Maximum number of iterations	50
Maximum number of generations	200
Transmission power (P_t)	[0.1 to 10] Watt
Receiver antenna gain	18 dBi
MHA gain (M_g)	2 dB
Cable loss (C_l)	2 dB
Noise figure (N)	2 dB
Body loss (L_b)	2 dB
Area	Urban
Maximum number of BSs	100
Longitude (upper-left X)	−100
Latitude (upper-left Y)	100
Longitude (lower-right X)	100
Latitude (lower-right Y)	−100
Number of users	10000
Propagation model	Cost-231 HATA Model

of next generation. The objective function in our approach can be formulated for getting the fitness (F) of the optimal network configuration as follows:

$$F = \left[\frac{UE^2}{T_P \times \text{ActiveBSs}^2} \right], \tag{5}$$

where UE is the number of connected users to the BSs, T_P is the total transmit power, and ActiveBSs is the number of BSs that connected to at least one user.

As for the termination criteria, we defined the maximum number of generations T_g. After executing simulation T_g generations, the proposed algorithm terminates the procedure and returns the best-so-far solution.

5. Experimental Results

In this section, the performance evaluation of the modified RGA is presented. At first, the modified algorithm is evaluated concerning the best-optimized power level and its location for 5G BSs with standard RGA and DE. In this scenario, we obtained the experimental results regarding the number of active base stations and transmission power with the connected users and compared with conventional DE. Finally, we also conducted the best, average, and worst fitness, transmission power, the number of active base stations, and their connected users of the modified RGA.

Table 2 lists the simulation parameters which we have considered in this paper. In the table, there are some constant variables such as bandwidth, carrier frequency, frame structure, receiver antenna gain, MHA gain, cable loss, noise figure, and body loss and decision variables such as population size (P), maximum number of iterations, and transmission power (Q). In our experiments, the environment area is assumed to be representing (X, Y) as $(-100.00, 100.00)$ and $(100.00, -100.00)$ in meter, where base stations and users are allocated in given area of interest. The users are allocated as an exact point by using their accuracy range in the given area as this is the new feature of 5G networks. Figure 3 shows a proposed simulation environment where we can see the following: rectangle boxes are representing cities with an area entirely covered with users mostly called urban area. The circle shapes are serving a coverage area by optimum base stations that were being proposed by the EA with the extended version of RGA.

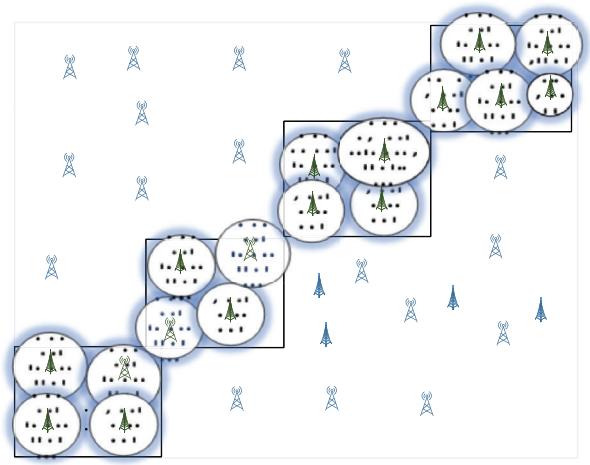


Figure 3: Simulation environment.

We have performed the experiments and reported values to estimate the best configuration for 5G BSs. Our experimental results gathered over 50 independent runs. The compared algorithms are described as follows:

(i) The modified RGA, with BSR = 0.1 and $M_r = 0.2$.

(ii) Standard DE(rand/1/bin) with SF = 0.5 and CR = 0.9.

(iii) Standard RGA with $M_r = 0.2$.

First of all, Figure 4 shows the convergence graphs of the modified RGA, DE, and RGA. As we can see, the modified RGA's performance is better than DE and RGA towards the upcoming generation. By comparing the algorithms (the modified RGA, DE, and RGA), we notice that RGA is getting slightly the same and worse fitness value towards a generation. This is because of shuffling happening again and again by using box crossover, where the modified RGA and DE are better than RGA as they are not changing their chromosomes every time in crossover operator.

Figure 5 shows that DE and RGA can keep less activated BSs and serve similar users at the same time in comparison with the modified RGA. This result leads to less fitness value as it depends on the objective function. The randomness of the EAs provides more of a chance for the network operator to find better BS combinations. However, this advantage comes at the expense of higher computational complexity that depends on the standard crossover and mutation of DE and RGA. That is why the modified RGA increases the number of the base stations more with less power consumption than DE and RGA to achieve better fitness.

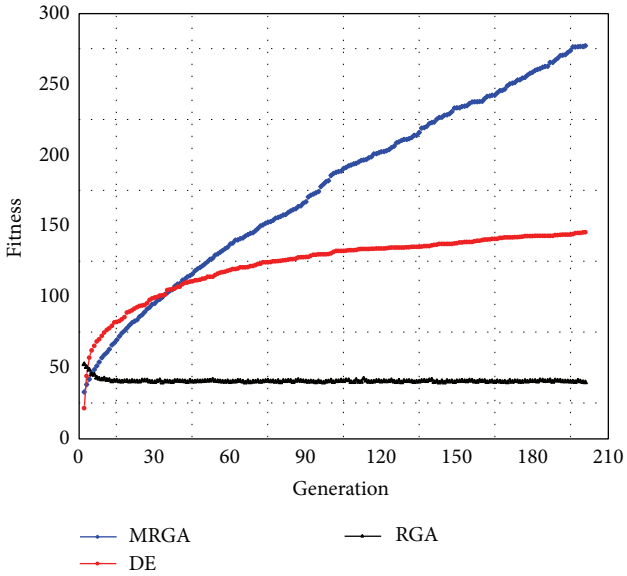


Figure 4: The convergence graphs.

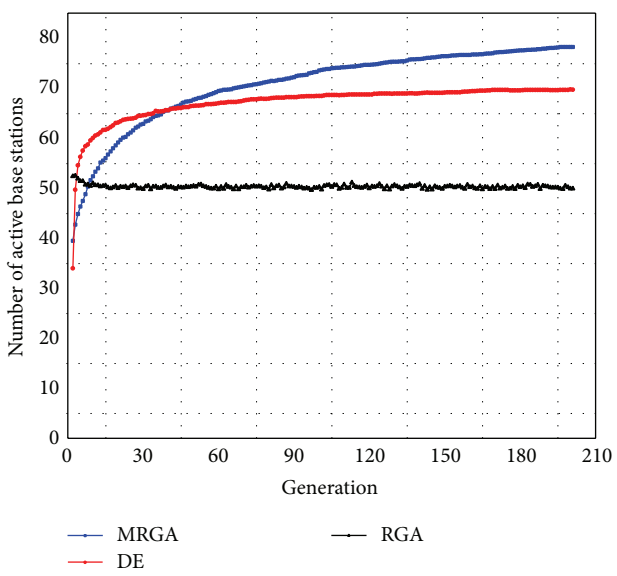


Figure 5: Comparison of the number of active BSs.

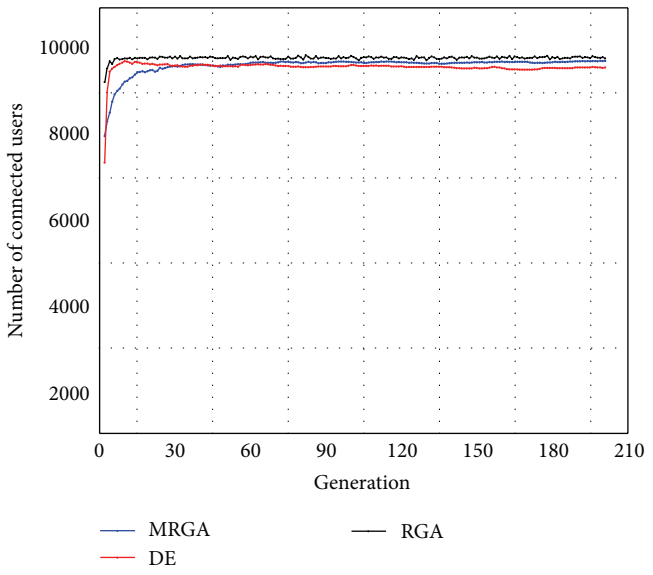


Figure 6: Comparison of the number of connected users.

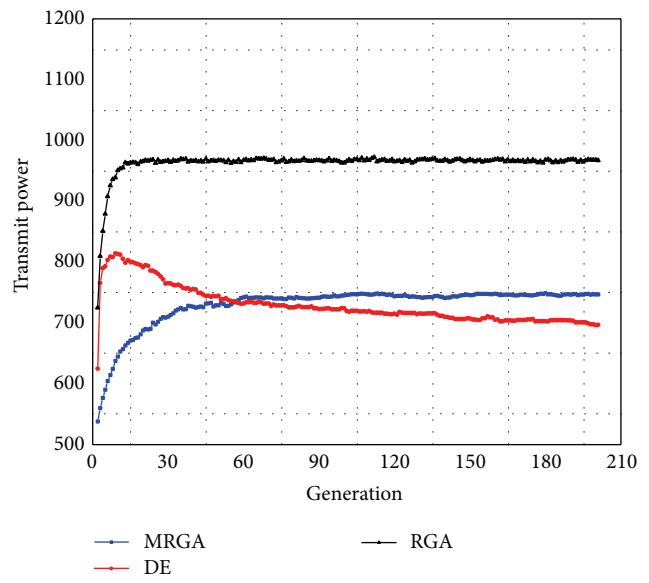


Figure 7: Comparison of the total transmit power.

In Figure 6, the most numbers of connected users are similar in the modified RGA, DE, and RGA, where we can see that coverage area is covered well for giving an excellent service to the connected users by using these three techniques. There is no much difference in performing to providing a good coverage in our simulation environment mentioned in Figure 3 by applying the modified RGA, DE, and RGA regarding the connected users.

Figure 7 plots the performance of transmission power consumption in between the modified RGA, DE, and RGA towards the upcoming generation, where we can see that DE and modified RGA are performing quite similarly after reaching around 60th generation due to replacing their chromosomes less than RGA. By doing less shuffling, the modified RGA and DE have performed well in terms of the power consumption. As a result, our goal of the mitigation of the energy consumption is reached by adjusting the BSs transmission power according to getting the optimal locations of BSs and covering the coverage area for satisfaction by the quality of service. In other words, our modified algorithm has less total power consumption in comparison to DE and RGA but still it can serve more users than DE and RGA, which makes the huge difference in their fitness.

At last, we have also gathered all the best, average, and worst results of our modified RGA approach. The results are plotted as fitness value in (a), transmission power in (b), a number of active base stations in (c), and the number of connected users in (d), respectively, in Figure 8.

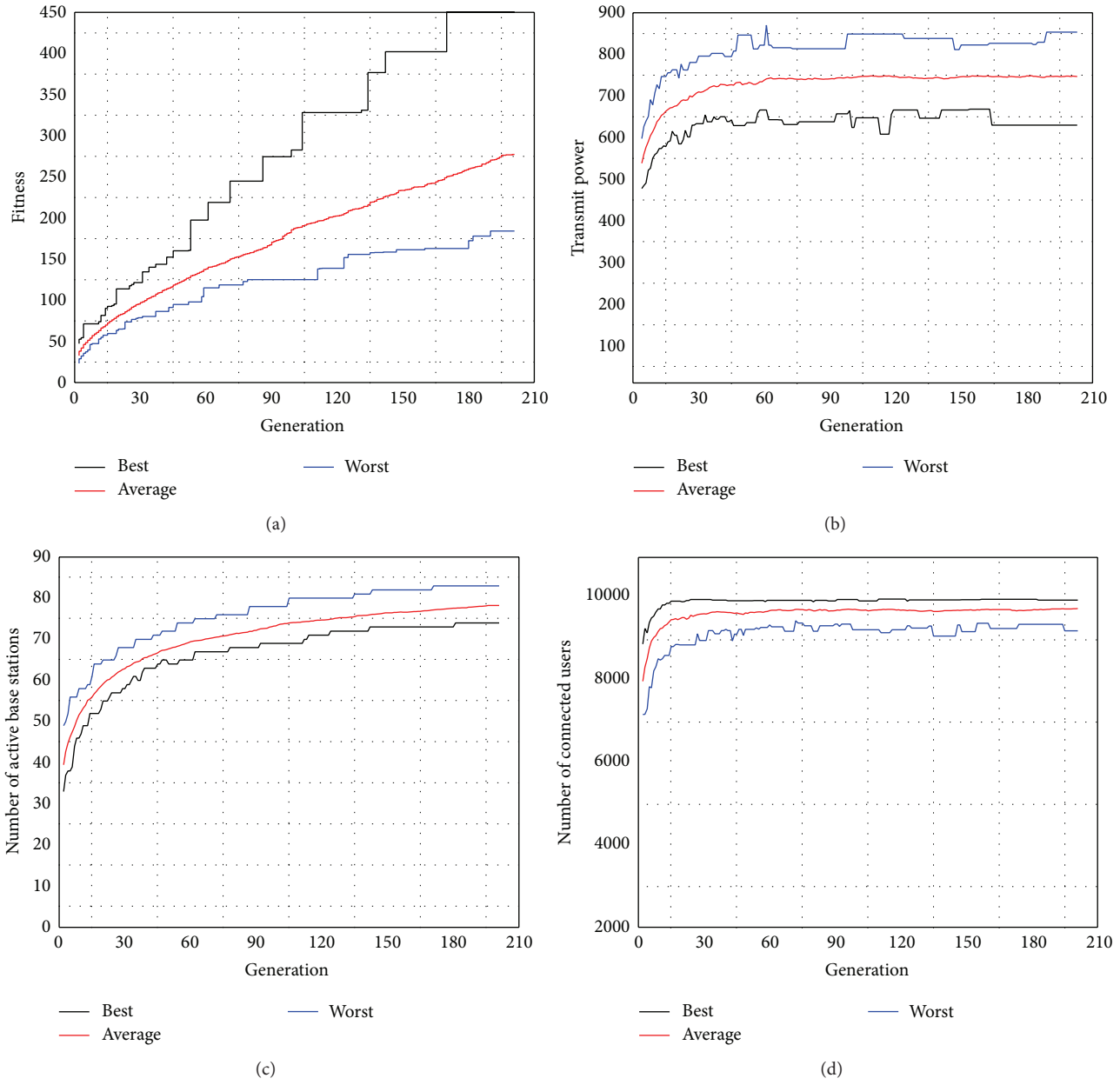


Figure 8: The best, average, and worst convergence graphs of the modified RGA.

Tables 3 and 4 show that our modified RGA is statistically significantly better than DE and RGA. As we can see in both of the tables, the t-value gives $2.01303E-22$ for DE and the modified RGA and $5.4306E-33$ for RGA and MRGA. Hence, it is proven that the modified RGA has statistically better performance than DE and RGA.

Table 3: The t-value of 49 degrees of freedom is significant at a QOS level of significance by two-tailed t-test for MRGA and DE.

	The modified RGA	DE	t-value
Average	277.270614	145.654808	
St. dev.	56.7086340687	15.8800909196	**$2.01303E-22$**

6. Conclusion

In this paper, we have offered the modified RGA method for allocating the optimal positions of the future 5G base stations. The modified RGA has achieved sufficiently better performance in terms of transmit power saving and total connected users for 5G networks with providing optimal coverage. We established to evaluate the location intelligence of the BSs to be in concession with green communications. The modified RGA has successfully found the considerable better configuration by comparing with conventional DE and RGA to locate proper location and adjusting the range of the power level additional to coverage constraints. In our ongoing and future

Table 4: The t-value of 49 degrees of freedom is significant at a QOS level of significance by two-tailed t-test for MRGA and RGA.

	The modified RGA	RGA	t-value
Average	277.270614	39.76453	
St. dev.	56.7086340687	2.399219965	**5.4306*E*−33**

work, we will study optimal BSs and their cost regarding frequency level using an advanced EA. We will also investigate the chronological evolution of the energy in the standard of satisfactory QoS.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This research was supported by X-Project funded by the Ministry of Science, ICT & Future Planning (NRF-2015R1A2A1A16074929).

References

[1] H. Holma and A. Toskala, *LTE for UMTS-OFDMA and SC-FDMA Based Radio Access*, John Wiley & Sons, New York, NY, USA, 2009.

[2] T. Han and N. Ansari, "Optimizing cell size for energy saving in cellular networks with hybrid energy supplies," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 5189–5193, IEEE, Anaheim, Calif, USA, December 2012.

[3] F. Boccardi, R. W. Heath Jr., A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[4] J. G. Andrews, S. Buzzi, W. Choi et al., "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[5] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5G networks: how location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 102–112, 2014.

[6] E. Amaldi, A. Capone, and F. Malucelli, "Radio planning and coverage optimization of 3G cellular networks," *Wireless Networks*, vol. 14, no. 4, pp. 435–447, 2008.

[7] S. Hurley, "Planning effective cellular mobile radio networks," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 243–253, 2002.

[8] Y. Liu, M. Tao, B. Li, and H. Shen, "Optimization framework and graph-based approach for relay-assisted bidirectional OFDMA cellular networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3490–3500, 2010.

[9] A. Ligeti and J. Zander, "Minimal cost coverage planning for single frequency networks," *IEEE Transactions on Broadcasting*, vol. 45, no. 1, pp. 78–87, 1999.

[10] S. Louvros, K. Aggelis, and A. Baltagiannis, "LTE cell coverage planning algorithm optimising uplink user cell throughput," in *Proceedings of the 11th International Conference on Telecommunications (ConTEL '11)*, pp. 51–58, IEEE, Graz, Austria, June 2011.

[11] V. Berrocal-Plaza, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez, "Artificial bee colony algorithm applied to WiMAX network planning problem," in *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications (ISDA '11)*, pp. 504–509, Córdoba, Spain, November 2011.

[12] A. Abdel Khalek, L. Al-Kanj, Z. Dawy, and G. Turkiyyah, "Optimization models and algorithms for joint uplink/downlink UMTS radio network planning with SIR-based power control," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1612–1625, 2011.

[13] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, "Load balancing and unbalancing for power and performance in cluster-based systems," in *Proceedings of the Workshop on Compilers and Operating Systems for Low Power*, vol. 180, pp. 182–195, Barcelona, Spain, 2001.

[14] A. Chatzipapas, S. Alouf, and V. Mancuso, "On the minimization of power consumption in base stations using on/off power amplifiers," in *Proceedings of the 1st IEEE Online Conference on Green Communications (GreenCom '11)*, pp. 18–23, IEEE, New York, NY, USA, September 2011.

[15] W. El-Beaino, A. M. El-Hajj, and Z. Dawy, "On radio network planning for next generation 5g networks: a case study," in *Proceedings of the International Conference on Communications, Signal Processing, and their Applications (ICCSPA '15)*, pp. 1–6, Sharjah, UAE, February 2015.

[16] H. G. Myung, *Technical Overview of 3GPP LTE*, Polytechnic University of New York, 2008.

[17] A. Al-Samawi, A. Sali, N. K. Noordin, M. Othman, and F. Hashim, "Base station location optimisation in LTE using genetic algorithm," in *Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC '13)*, pp. 336–341, IEEE, Jeju, Republic of Korea, October 2013.

[18] S. Sakthivel and R. Suresh, "A genetic algorithm approach to solve mobile base station location problem," *International Journal of Soft Computing*, vol. 1, no. 3, pp. 160–165, 2006.

[19] B. Jalili and M. Dianati, "Application of taboo search and genetic algorithm in planning and optimization of UMTS radio networks," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference (IWCMC '10)*, pp. 143–147, ACM, July 2010.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2016, Article ID 5201937, 11 pages
http://dx.doi.org/10.1155/2016/5201937

Research Article

A Hybrid IP/GA Approach to the Parallel Production Lines Scheduling Problem

Huizhi Ren[1] and Shenshen Sun[2]

[1]*School of Mechanical Engineering, Shenyang University of Technology, Shenyang 110870, China*
[2]*School of Information Engineering, Shenyang University, Shenyang 110044, China*

Correspondence should be addressed to Huizhi Ren; renhuizhi@126.com

Received 11 December 2015; Revised 12 March 2016; Accepted 14 April 2016

Academic Editor: Yourim Yoon

A special parallel production lines scheduling problem is studied in this paper. Considering the time window and technical constraints, a mixed integer linear programming (MILP) model is formulated for the problem. A few valid inequalities are deduced and a hybrid mixed integer linear programming/constraint programming (MILP/CP) decomposition strategy is introduced. Based on them, a hybrid integer programming/genetic algorithm (IP/GA) approach is proposed to solve the problem. At last, the numerical experiments demonstrate that the proposed solution approach is effective and efficient.

1. Introduction

Electric wire and cables are usually continuously produced and can be in coil packaging (or circle packing) [1]. In the large cable manufacturing enterprise, there is usually more than one continuous cable production line and cables can be produced on most of the lines. For the same cable product, the different lines always take different production cost and different production time. It raises a parallel scheduling problem over these production lines. The objective is to minimize the total production cost. The due time of each cable product and technological requirements of each cable production line need to be satisfied. The problem can be viewed as a parallel machine scheduling problem with complicated constraints of continuous cable production lines. It includes two kinds of scheduling decisions: a job allocation decision over multiple machines (cable production lines) and a job sequencing decision on each machine (cable production line).

The parallel machine scheduling problem is a common combinatorial optimization problem and there have been a lot of related researches published. In the 1970s, a series of researches have ever focused on problem complexity analysis, as shown in Table 1, where the symbol description system in Graham et al. [2] was adopted and the identical parallel machine problems with various objectives in literature [3–6] were proven to be NP-hard while the heterogeneous parallel machine scheduling problem with total complete times was provided an algorithm with $O(n \log n)$ complexity. Later, Dessouky et al. [7] also proved that heterogeneous parallel machine problems, including Q/r_i, $p_i = 1/C_{\max}$, $Q/p_i = 1/\sum \omega_i c_i$, $Q/p_i = 1/L_{\max}$, and $Q/p_i = 1/\sum T_i$, have $O(n \log n)$ complexity. Various kinds of algorithms have been developed to solve the parallel machine scheduling problems of NP-hardness, including the intelligent optimization algorithms, such as Tabu Search [8–10], Simulated Annealing [11], and genetic algorithm [12, 13] and optimization algorithms, such as Branch and Bound [14–17], Dynamic Programming [18–20], and Column Generation [21–23].

It is worth noting that a hybrid mixed integer linear programming/constraint programming (MILP/CP) approach has been proposed to solve a class of parallel machine scheduling problems (Harjunkoski et al. [24], Jain and Grossmann [25], Hooker [26], and Maravelias and Grossmann [27]) which are similar to the problem studied in this paper. The approach divides the parallel scheduling problem into a job allocation master problem and a series of single machine scheduling subproblems. It applies the mixed integer linear programming (MILP) to solve the former master problem and constraint programming (CP) to the latter subproblems. Typically, the process needs to be iterated because

Table 1: The complexity research of various parallel machine problems in the 1970s.

Literature	Year	Problem variants	Complexity
Karp [3]	1972	$P2//C_{\max}$	NP-hard
Bruno et al. [4]	1974	$P2//\sum \omega_i c_i$	NP-hard
Lenstra [5]	1977	$I/r_i/\sum c_i$	NP-hard
Garey and Johnson [6]	1978	$P//C_{\max}$	NP-complete
Horowitz and Sahni [29]	1976	$Q//\sum c_i$	$O(n \log n)$

the CP always encounters an infeasibility in solving the single machine scheduling subproblem. The hybrid MILP/CP approach can obtain an optimal solution after a number of iterations. Its limitation is the computational inefficiency. It can deal with only small-scale problems in an acceptable CPU time, although some improvements have been developed [27, 28] to reduce the number of iterations.

The hybrid MILP/CP approach can also be applied to the parallel cable production line scheduling problem after revising its CP model according to the particular scheduling requirement of the cable production line. The hybrid MILP/CP approach, though dealing with only some small-scale problems, motivates us to combine the integer programming (IP) and genetic algorithm (GA) together to develop the hybrid IP/GA approach to the parallel cable production lines scheduling problem. In the hybrid IP/GA approach, the traditional exact optimization method (IP) and the intelligent optimization method (GA) can complement each other to strengthen the solving process, similar to what the MILP and CP perform in the hybrid MILP/CP approach. This is our main contribution in the paper.

The outline of this paper is as follows. Section 2 describes constraints of the problem and provides its integer programming model. In Section 3, a few valid inequalities are deduced and a hybrid MILP/CP model is presented. In Section 4, a hybrid IP/GA approach is proposed to resolve the problem. Section 5 tests the proposed solution approach. Section 6 concludes the paper.

2. Problem Description and Mathematical Model

2.1. The Problem Description. In the cable manufacturing enterprise of our own interest, cables are produced coil by coil through the equipped production lines and thus a cable coil, hereinafter referred to as the "cable," will be viewed as a job to be scheduled. The cable requires different processing time and cost on different production lines because the lines are configured with different product preferences. The processing time, including the setup time, is assumed to be independent of processing sequence and thus the setup time will be ignored (or included in the processing time) in this problem. The scheduling objective is to minimize the total processing cost.

There are always some cables requiring more cable rubber and the production line needs to provide more heated rubber for these cables, which will form a technical constraint for the scheduling problem. To facilitate the problem description, the cables that need more rubber will be classified into "type B" and others into "type A." The cables of type B cannot be processed consecutively in a production line and several cables of type A are needed between processing two successive cables of type B; otherwise the cable rubber quality could be impaired.

Another constraint comes from the time window of each cable. Since the cable cores are handed off from the upstream process, each coil has a release time in the cable production line. In addition, most cables have the contracted delivery time. These two times constitute the time window constraint.

A cable production scheduling problem instance is illustrated in Figure 1. It needs to schedule seven cables (or jobs, labeled as $J1, J2, \ldots, J7$ in Figure 1) over two production lines (labeled as $L1$ and $L2$). Cables $J1$ and $J2$ belong to type B and others belong to type A. As shown in the top part of Figure 1, each cable has its own time window $[r_i, d_i]$ and two different processing times, p_1^1 and p_1^2, corresponding to the two production lines, respectively. A schedule arrangement is shown in the bottom part of Figure 1. For the time window constraint and technical requirement (forbidding cables $J1$ and $J2$ to be processed continuously on line $L1$ or $L2$), no production line processes cable $J2$ and thus the schedule is infeasible.

2.2. The Mixed Integer Linear Programming Model. To define and formulate the problem explicitly, the following notations need to be introduced to represent all coils, production lines, coil type parameters, and so on:

N: set of cables and $N = \{0, 1, 2, \ldots, n\}$, where 0 is a dummy cable.

M: set of cable production lines, $M = \{1, 2, \ldots, m\}$.

B: set of cables belonging to type B.

A: set of cables belonging to type A, $A \cap B = \emptyset$, $A \cup B = N$.

Q: number of necessary cables of type A between successive cables of type B in a production line.

r_i: release time of cable i, $i \in N$.

d_i: due time of cable i, $i \in N$.

c_{ij}: production cost of cable i on line j.

p_{ij}: production time of cable i produced on line j.

Z: a sufficiently large constant.

To represent cable assignment decisions over the production lines and scheduling decisions on each line, we introduce the following decision variables:

$$y_{ij} = \begin{cases} 0, & \text{if cable } i \text{ is produced on line } j; \\ 1, & \text{otherwise.} \end{cases}$$

$$x_{ii'j} = \begin{cases} 0, & \text{if cable } i' \text{ is produced immediately after } i \text{ on line } j; \\ 1, & \text{otherwise.} \end{cases}$$

S_i: start time of producing cable i, $i \in N$.

C_i: complete time of cable i, $i \in N$.

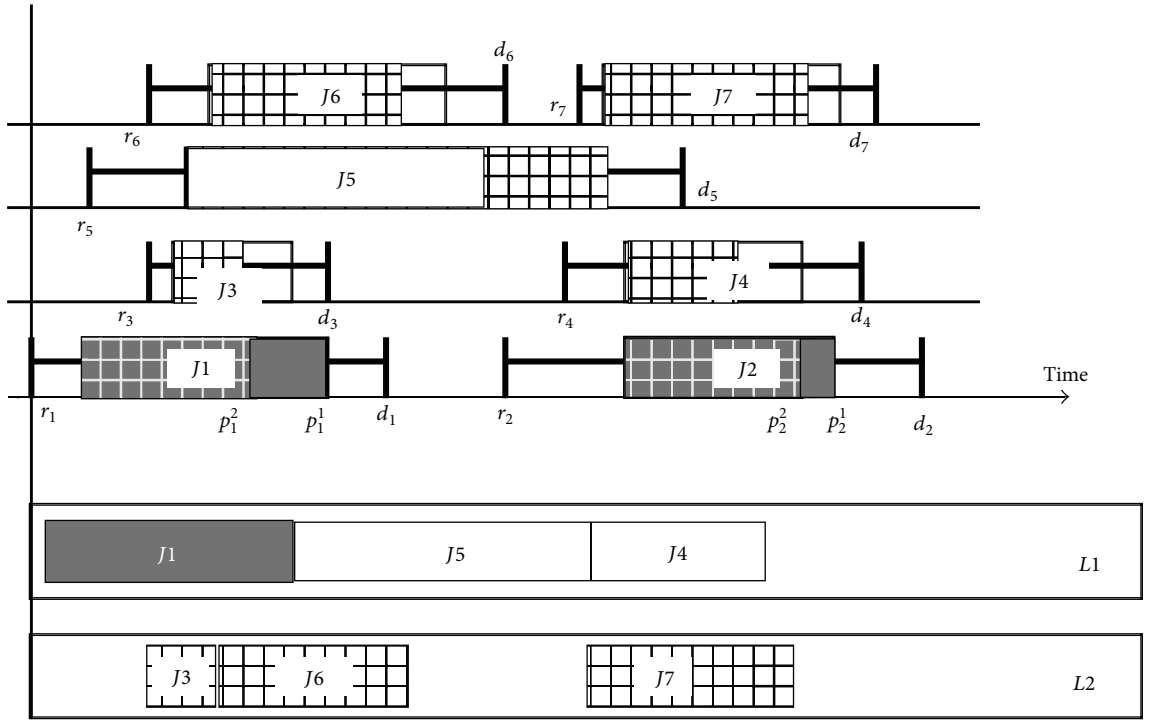


Figure 1: An illustration of the cable parallel production lines scheduling problem.

The following auxiliary variables are also introduced to represent the number of consecutive cables of type A on a production line, which is necessary to represent the technological requirement of production lines:

a_i: number of consecutive cables of type A before cable i, $i \in A$.

Based on the above notations, the parallel cable production lines scheduling problem can be formulated as the following mixed integer linear programming model:

$$\text{Min} \quad \sum_{i \in N \setminus \{0\}} \sum_{j \in M} c_{ij} y_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in M} y_{ij} = 1, \quad i \in N \setminus \{0\} \tag{2}$$

$$\sum_{i \in N, i \neq i'} x_{ii'j} = y_{i'j}, \quad i' \in N \setminus \{0\}, \ j \in M \tag{3}$$

$$\sum_{i' \in N, i' \neq i} x_{ii'j} = y_{ij}, \quad i \in N \setminus \{0\}, \ j \in M \tag{4}$$

$$\sum_{i' \in N \setminus \{0\}} x_{0i'j} \leq 1, \quad j \in M \tag{5}$$

$$a_{i'} \leq Z \left(1 - \sum_{j \in M} \sum_{i \in B} x_{ii'j} \right), \quad i' \in A \tag{6}$$

$$a_{i'} \leq a_i + 1 + Z \left(1 - \sum_{j \in M} x_{ii'j} \right), \quad i, i' \in A \tag{7}$$

$$\sum_{j \in M} x_{ii'j} \leq 0, \quad i, i' \in B \tag{8}$$

$$a_i \geq Q - 1 - Z \left(1 - \sum_{j \in M} x_{ii'j} \right), \quad i \in A, \ i' \in B \tag{9}$$

$$S_i \geq r_i, \quad i \in N \setminus \{0\} \tag{10}$$

$$C_i \leq d_i, \quad i \in N \setminus \{0\} \tag{11}$$

$$S_{i'} - C_i \geq Z \left(\sum_{j \in M} x_{ii'j} - 1 \right), \quad i, i' \in N \setminus \{0\} \tag{12}$$

$$C_i - S_i \geq \sum_{j \in M} p_{ij} y_{ij}, \quad i \in N \setminus \{0\} \tag{13}$$

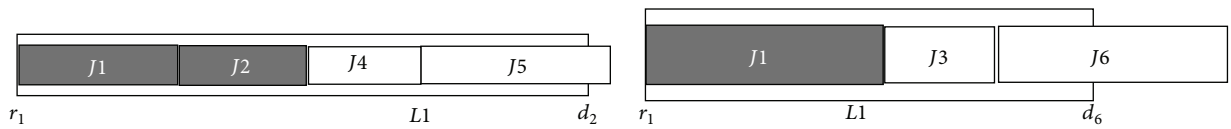
$$x_{ii'j}, y_{ij} \in \{0, 1\}, \quad i, i' \in N, \ j \in M \tag{14}$$

$$S_i, C_i \geq 0, \quad i \in N \tag{15}$$

$$a_i \in \{0, 1, 2, \ldots, n\}, \quad i \in A. \tag{16}$$

The objective function (1) of the model is to minimize the total production cost of the adopted cable production tasks assignments. Constraints (2)–(5) state the logic relations among the decision variables. Constraint (2) ensures that a production line is assigned to each cable. Constraints (3) and (4) ensure that each cable has one preceding cable and one succeeding cable, respectively. Constraint (5) states that each line has at most one starting cable, which implies that there are at most m available lines.

Constraints (6)–(9) aim to the technical requirement that there are enough cables of type A between any two successive cables of type B on the same line. Among them, constraint (6) states that $a_{i'} \leq 0$ if its preceding cable belongs to type B.



(a) A cable assignment against the production capacity inequality (b) An assignment against the production time equilibrium inequality

Figure 2: Cable assignments excluded by the valid inequalities.

Constraint (7) calculates the number of consecutive cables of type A. Constraint (8) prevents the consecutive cables of type B while constraint (9) guarantees no less than Q cables of type A between any two cables on the same line.

Constraints (10) and (11) are the time window constraints. Constraint (12) ensures the cable production sequence on the same line and constraint (13) provides enough production time for each assignment. Constraints (14)–(16) provide value ranges for the variables.

3. Valid Inequalities, a Hybrid MILP/CP Solution Method, and Lower Bound for the Problem

3.1. Valid Inequalities. In this section, a few valid inequalities will be introduced. To the problem definition, MILP model (1)–(16) is already appropriate and all the valid inequalities are redundant, but they can reduce the computing time if we attempt to directly solve the model through optimization software. It is more important that the valid inequalities can reflect the problem features and work for the heuristic approach.

Production Capacity Inequality. Consider the following:

$$\sum_{i \in N} y_{ij} p_{ij} \leq \operatorname{Max}\{d_i \mid i \in N\} - \operatorname{Min}\{r_i \mid i \in N\}, \quad j \in M. \tag{17}$$

Inequality (17) prevents total production times of assignments to any line more than its capacity in the scheduling horizon. For the case in Figure 1, cables $J1$, $J2$, $J4$, and $J5$ cannot be assigned to line $L1$ concurrently because inequality $p_{11} + p_{21} + p_{41} + p_{51} > \operatorname{Max}\{d_i \mid i \in N\} - \operatorname{Min}\{r_i \mid i \in N\} = d_2 - r_1$ stands, which conflicts with inequality (17), as shown in Figure 2(a).

There is still a tighter inequality which takes into consideration the production tasks balancing in time.

Production Time Equilibrium Inequality. Consider the following:

$$\sum_{i \in \Psi(r_{i'}, d_{i''})} y_{ij} p_{ij} \leq d_{i''} - r_{i'}, \quad i, i' \in N,\ d_{i''} > r_{i'},\ j \in M, \tag{18}$$

where $\Psi(r_{i'}, d_{i''}) = \{i \mid r_i \geq r_{i'} \text{ and } d_i \leq d_{i''},\ i \in N\}$ includes all cables that need to be produced in time window $[r_{i'}, d_{i''}]$.

Inequality (17) can be viewed as a special case of inequality (18) and inequality (18) is a tighter inequality. For the case illustrated in Figure 1, $\Psi(r_1, d_6) = \{1, 3, 6\}$ and cable combination $J1$, $J3$, and $J6$ can be excluded from $L1$ through inequality (18), although the cable combination meets inequality (17), as shown in Figure 2(b).

It is worth noting that inequality (18) is not adequate to ensure the production equilibrium absolutely in time because the real production duration is less than the given time window and permitted to slide in the time windows. Therefore, the assignment scheme satisfying inequality (18) can still be infeasible for one or more productions even if ignoring the technical requirements (see [24, 28] for more details). In addition, inequality (18) can be significantly tightened again as in Ren and Liu [28].

Limit of Cable Rate of Type B in a Production Line. In effect, the technical constraint (9) limits the ratio of the cable numbers of type B and type A in a production line:

$$\sum_{i \in A} y_{ij} \geq (Q - 1) \sum_{i \in B} y_{ij}, \quad j \in M. \tag{19}$$

Inequality (19) provides a filtering mechanism for the cable assignment over the production lines. For the assignment in Figure 2(a), cable combination $J1$, $J2$, and $J4$ can be filtered out from any line when $Q \geq 2$, because at least two cables of type A are necessary between cables $J1$ and $J2$.

3.2. Hybrid MILP/CP Approach and Lower Bound for the Problem. In the hybrid MILP/CP approach [24, 25], the parallel cable production lines scheduling problem should be decomposed into a relaxed master problem, focusing on the cable assignment decision, and a single production line scheduling subproblem. The relaxed master problem can be formulated into an IP model by formulae (1), (2), and (18). The subproblem for a single production line is formulated by the following CP model:

$$\text{All-different}(W); \tag{20}$$

$$\text{if } (W_k \in B) \quad \left(b_{W_{k+1}} + b_{W_{k+2}} + \cdots + b_{W_{k+Q}} = Q\right) \vee (L < (k + Q)) \tag{21}$$

$$\text{The required production time and time window constraints.} \tag{22}$$

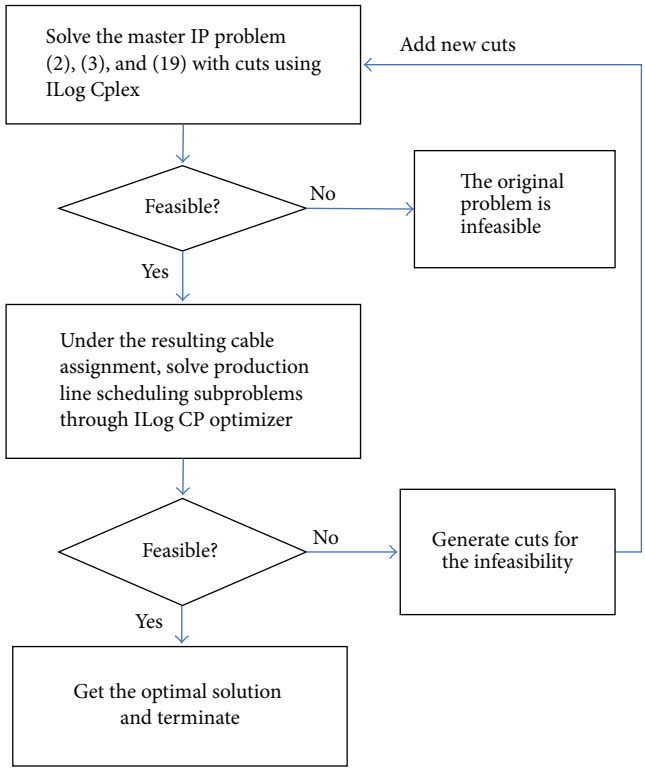


Figure 3: Illustration of the hybrid MILP/CP approach.

In the above CP model, W_k denotes the ith cable on the considered production line and L is the number of cables assigned to the line. Formula (20) means that the L cables are all different from each other. Formula (21) is the technical constraint in the line which requires at least Q cables of type A immediately after a cable of type B.

As illustrated in Figure 3, the hybrid MILP/CP approach begins with solving the relaxed master problem through solving the corresponding IP model optimally. Here, the IP model is solved by ILOG Cplex, a commercial optimization software package. Based on the optimal solution to the master problem, the cable set for each production line can be determined temporarily. The CP model (20)–(22), solved by ILOG Scheduler (an ILOG CP Optimizer), can deal with the m single line scheduling subproblems. If the m subproblems are all feasible and m single line schedules are obtained, the computing process terminates and the optimal solution is obtained. Otherwise, a "cut," a kind of constraint to eliminate the current infeasible cable assignments (as formulated by inequality (23)), will be constructed and appended to the master problem model which will trigger a new repeat. The iterated process will repeat until all the m feasible single line schedules are obtained.

The "cut" in the iterating process can be formulated as

$$\sum_{i \in \Omega} y_{ij^*} \leq \|\Omega\| - 1, \tag{23}$$

where j^* is the production line encountering infeasibility and Ω is a set of cables that contribute to the infeasibility.

The hybrid approach, with improvement in Ren and Liu [28], can solve only small-scale problems, even if ignoring the special technical constraints. For the problem instances of larger size in this paper, it can provide a lower bound and the lower bound can be increased with the more cuts in iterations. But the iterating process can become very time consuming and the lower bound always keeps fixed in many iterations for large problem sizes. Another significance of the hybrid approach is to give an initial cable assignment scheme. Although it is typically infeasible, our heuristic solution can view it as a starting point of the search process.

4. Hybrid IP/GA Approach to the Problem

Genetic algorithm (GA) is a metaheuristic based on evolution population and explores multiple solutions concurrently and thus performs well at global search which is complementary for the optimality (with infeasibility) and solution directivity of the integer programming (IP) method. More GA details can be found in Goldberg [30], Grefenstette [31], and Oliveto and Witt [32]. There are some published studies of genetic algorithm applications in kinds of scheduling problems [33–37]. Through evolution, they find high-quality solutions within reasonable computing time.

Generally, a desirable GA can converge quickly at the same time avoiding premature convergence to low-quality solutions. In practice, it needs a good balance between computational time and the solution quality [32]. To improve the GA searching process, an optimization approach, the integer programming (IP) method based on the IP model (1), (2), and (18), is introduced to the iterating steps. It can not only improve the GA searching process but also provide a better lower bound to evaluate the final solution quality.

4.1. Framework of the Hybrid IP/GA. The hybrid IP/GA approach is illustrated in Figure 4. It is characterized by the interaction between IP and GA in the iterating process. The optimization method with IP model provides cable assignment schemes to the GA and the GA feeds back to the IP model some "cuts." These "cuts" can cut off some cable assignment schemes and tighten the IP model in turn. The tighter model can possibly produce better solution (cable assignment scheme) and better lower bound.

4.1.1. Generating Cable Assignment Schemes through IP. As mentioned in Section 3.2, it can provide a cable assignment scheme to solve IP model (1), (2), and (18) through the optimization software (such as ILOG Cplex). To satisfy the technical constraint, inequality (19) can also be inserted into the IP model. The cable assignment scheme can be extended to a solution through applying the earliest due date (EDD) rule to each production line. The solution, which violates one or more of the time window constraints in most cases but is "optimal" for the objective function, can be taken as a seed to generate the initial chromosomes for GA. The infeasibility can be accepted for chromosome but causes a great penalty in the fitness function.

In the evolution process of GA, the IP model can be attached with a series of "cuts," inequalities as formula (23),

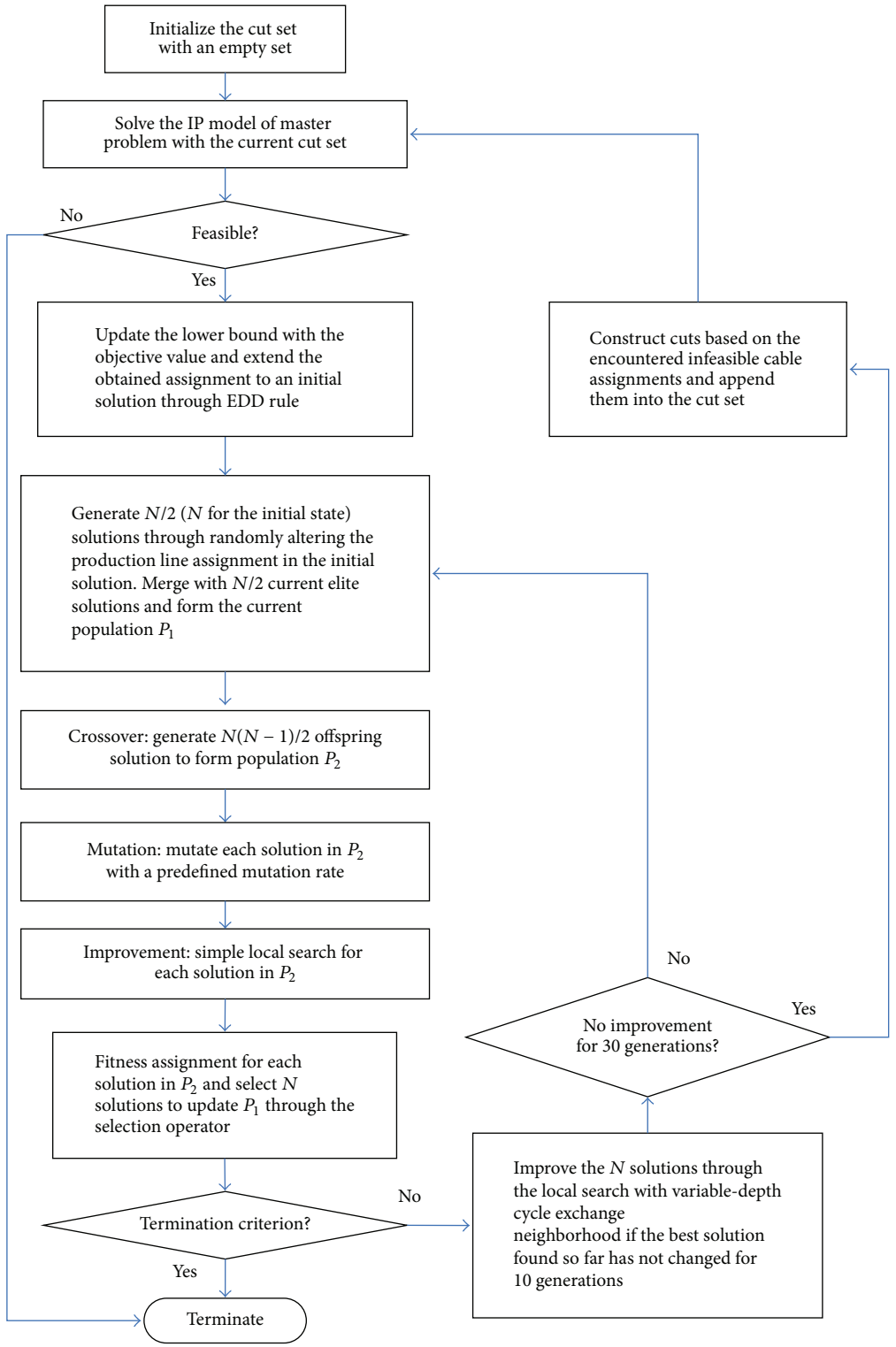


Figure 4: The algorithm framework of the hybrid IP/GA approach.

and generate more diverse cable assignment schemes. The cable assignment schemes can also be extended to solutions and provide diverse chromosomes to the GA. In our hybrid algorithm, the IP model solver is called for a certain number of generations.

4.1.2. Generating Cuts for IP in the GA Generations. Some of the chromosomes and their infeasibility information in the GA generations need to be retained and form "cuts" to be appended to the IP model. To generate new cable assignment schemes, the cut set of IP model needs to be updated each

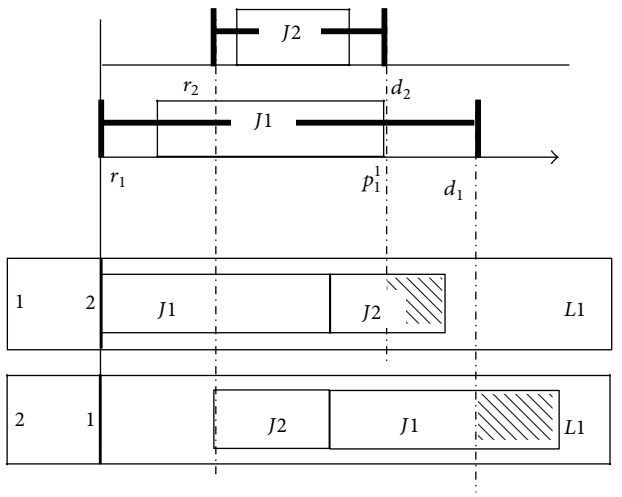


Figure 5: The "cut" for an infeasible cable assignment.

time the IP solver is called; otherwise, it will generate the same scheme as the last.

An infeasibility and the "cut" are illustrated in Figure 5. It is infeasible to assign cables $J1$ and $J2$ to line $L1$, irrespective of the processing sequence, due to the nonpreemption and time window constraint, although it satisfies inequality (18). Based on formula (23), a "cut," formulated as inequality $y_{11} + y_{21} \leq 1$, can be appended into the IP model of the master problem to eliminate the infeasible cable assignment.

It is worth noting that the infeasibility needs to be verified through CP model in Section 3.2 because our single machine scheduling process is not an exact algorithm and cannot guarantee a feasible schedule for each feasible cable combination on a single production line although it does in most cases. If the infeasibility is denied by the CP solver, that is, a feasible schedule is outputted, the cable combination assigned to the line should be accepted and the related solution should be a feasible solution. In the numeric tests, the presumptive case has never happened. On the other hand, some feasible cable assignments can also be used to form "cuts" to make the initial cable assignments more diverse.

Because the CP solver can waste more computing time and too many cuts can decrease the solving speed of the IP model, the infeasible cable combinations (or cuts) need to be selected carefully. In the hybrid approach, only cuts from the assignment schemes with currently optimal objectives can be selected to the CP model.

Typically, IP model (i.e., the relaxed master problem) with more cuts has more chance to improve the lower bound for the problem.

4.2. Genetic Coding for the Problem. To facilitate the crossover and mutter operators, the vector-group coding method [38] is adopted. In the coding method, a chromosome is constructed using a sequence of two-dimensional vectors as follows:

$$\mathrm{Ch} = \begin{bmatrix} g_1 & g_2 & \cdots & g_n \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & \cdots & h_n \\ k_1 & k_2 & \cdots & k_n \end{bmatrix}, \quad (24)$$

where $g_i = \begin{bmatrix} h_i \\ k_i \end{bmatrix}$ represents the ith genein chromosome Ch, $h_i \in N$ is a cable, $k_i \in M$ is production line to process cable h_i, and $\{h_1, h_2, \ldots, h_k\} = N \setminus \{0\}$. The cables assigned to the same line will be processed in gene order of the chromosome.

The start time of each cable production is equal to the maximum of the complete time of the previous cable and its own release time.

4.3. Fitness Function and Selection Operator. Since the objective function needs to be minimized and some chromosome can be infeasible, we define a new referred function by

$$f'(s) = f(s) + N_s P, \quad (25)$$

where $f(s)$ is the objective function value of chromosome s, N_s is the number of cables that violate their respective time windows or technical constraint (9), and P is a predefined constant larger than the expected value of $f(s)$.

We use a common selection operator: roulette-wheel selection. A dynamic linear fitness function [30] $U_t(s) = 1.1 \max\{f'(s) \mid s \in P(t)\} - f'(s)$, where t is the generation number of the current population $P(t)$.

4.4. Crossover Operator. Based on the vector-group coded chromosomes, we adopt one-point and two-point extended order crossovers. Let two chromosomes, $\mathrm{Ch} = [g_1 \ g_2 \ \cdots \ g_n]$ and $\mathrm{Ch}' = [g'_1 \ g'_2 \ \cdots \ g'_n]$, serve as the present chromosomes. The following describes how to cross over to produce new chromosome.

In the one-point extended order crossover, a random integer λ with $1 \leq \lambda < m$ is first generated to decide the crossover position. New chromosome $\mathrm{Ch}'' = [g_1 \ g_2 \ \cdots \ g_\lambda \ g'_{\lambda+1} \ \cdots \ g'_n]$ if $\{g_1, g_2, \ldots, g_\lambda\}$ and $\{g'_{\lambda+1}, \ldots, g'_n\}$ do not include a common cable; otherwise, discard genes with common cable in Ch' and insert the genes corresponding to the lost cables in Ch', since the number of common cables must be equal to the number of lost ones. It can generate different chromosome to reverse the Ch and Ch'.

In the two-point extended order crossover, two random integers λ_1 and λ_2 with $1 \leq \lambda_1 < \lambda_2 < m$ are first generated to decide the crossover positions. New chromosome $\mathrm{Ch}'' = [g'_1 \ g'_2 \ \cdots \ g_{\lambda_1} \ g_{\lambda_1+1} \ g_{\lambda_2} \ \cdots \ g'_{\lambda_2+1} \ \cdots \ g'_n]$ if no replicate cable is involved in the crossover. The treatment for the replicate cable is the same as that in the one-point crossover.

4.5. Mutation Operator. The bit-mutation and swap-mutation are adopted. In view of the signification of line assignment decisions for cables, the bit-mutation mutates only the second row of the chromosome genes, replacing it with a random number from $\{1, 2, \ldots, m\}$. It changes the production line to process the cable.

The swap-mutation swaps randomly selected gene values in a chromosome. It can change only the production sequence of one or two production lines. The technical and time window constraints can be involved in the mutation.

4.6. Local Search. To improve the evolution process, a simple local search, based on the swap and insert neighborhoods,

is applied to all chromosomes in the population for all generations while a local search, based on a variable-depth cycle exchange neighborhood (see Luo and Tang [39] for more details), is applied to chromosomes when the best solution found so far has not changed for 10 generations.

The swap neighborhood is obtained through exchanging the assigned production lines of each pair of genes in the given chromosome. The production orders should be exchanged if the pair of genes is corresponding to the same line. The processing sequence should be changed locally according to the technical constraint (formula (9)) and EDD rule which can improve the solution, if the production lines are exchanged. Similar to the swap neighborhood, the insert neighborhood involves inserting a cable to another line or a different position in the same line. The technical constraint has a higher priority than the EDD ruler when they conflict.

5. Computational Experiments

To test the performance of the formulated mathematical model and the proposed heuristic approach with improvement strategies, we complemented all the involved programs under the development environment of VC++ 2010 and solved the IP model and CP with *ILog Cplex* and *ILog Scheduler*, respectively. The experiments were all performed on a computer with Win 7 operating system, 2.8 Ghz Intel 2 Core CPU, and 4 GB RAM.

5.1. Configuration of the Hybrid IP/GA Approach. GA parameters for all problems were set to be the same: population size = 100, crossover probability = 1, mutation probability = 0.1, and the algorithm stops when gap between the lower bound and the best solution found so far has not been improved for 60 generations, the gap is lower than 0.5%, or the total number of generations reaches 600 (which comes from the computing time limit). The IP master problem was triggered when the best solution found so far has not been improved for 30 generations.

5.2. Test Problem Instances. We tested 12 problem instances in the numerical experiments from a practical cable production enterprise, which had five production lines but only four of them were available in these instances. The production cost and times were both estimated based on the empirical data. The number of necessary cables of type A between successive cables of type B in a production line was three and thus $Q = 3$ in all test problems and the proportion of type B was less than 20% in practice.

To test the developed hybrid IP/GA approach thoroughly, a series of problem instances of six different sizes were generated through imitating the practical data. Each problem size has 12 instances. The processing costs (c_{ij} for cable i and production line j) were random integers from uniform distribution of $U[20, 100]$, which magnified the practical figures about 10 times for more comparability. The processing times were from $U[30, 60]$. The release times (r_i) were from $U[0, T_{\max}]$ and due times $d_i = r_i + \delta p_i$, where $T_{\max}$ should be large enough to ensure the feasibility, $p_i = \text{Max}\{p_{ij} \mid j \in M\}$ is the maximal processing time of the cable i, and δ is a magnification factor randomly from $U[1.5, 3.5]$.

Table 2: Experimental result for the small sized problems.

Problem	$n \times m$	Objective value		CPU time (seconds)	
		MILP/CP	IP/GA	MILP/CP	IP/GA
1	12×2	56	56	3.41	<1
2	12×2	58	58	0.86	<1
3	16×2	61	61	1.14	1.13
4	16×2	79	79	14.87	1.21
5	15×3	81	81	3.48	1.79
6	15×3	67	*67**	143.34	2.76
7	18×3	94	94	23.55	4.84
8	18×3	87	87	6.56	4.16
9	16×4	67	67	31.42	2.89
10	16×4	53	53	39.38	4.54
11	20×4	102	*102**	207.51	5.30
12	20×4	113	113	71.35	6.13

*The optimal solution is obtained but gap is not equal to 0.

5.3. Computational Results. The small sized problems were first solved to optimality using the hybrid MILP/CP algorithm and then solved by the proposed hybrid IP/GA. The small sized problem instances were cut out from the 12 practical instances. $n \times m$ represented the problem size, where n and m denote the number of cables and production lines, respectively. Table 2 shows the experimental results with the different problem sizes. For the 12 problems of small size, the hybrid IP/GA also obtained almost all the optimal solutions but did not verify the optimality for two of them.

The experimental results for the 12 practical problems are listed in Table 3. Each problem has four production lines and the problem size is represented by the number of cables. In the table, Gap = (Objective value−Lower bound)/Lower bound is used to evaluate the solution quality. The average relative gap was 2.78% and the average CPU-s was 83.

For the six sets of randomly generated problem instances, Table 4 lists the average solution gap and CPU-seconds. The average relative gap was 2.51% and the average computing time was 121 CPU-seconds. In addition, the number of iterations reaches up to the limit (600 generations) for most of the instances.

To further test the performance of the hybrid IP/GA approach, we also solved the problems with a GA by Vallada and Ruiz [40], which was proposed to solve unrelated parallel machine scheduling problem sharing similarities with our problem. Figure 6 shows the comparison results of the average relative objective value (ARD) of the group of practical problem instances and six sets of randomly generated instances. From the results, the efficiency of our algorithm in solving our problem has been verified. In addition, our proposed hybrid IP/GA approach can provide a better lower bound for the problem.

5.4. Computational Analysis. From the computational results, all the problem instances can be resolved in

Table 3: Experimental results for the practical problems.

Problem	Size	Objective value		Gap (%)	CPU time (seconds)
		IP/GA	LB		
1	41	172	170	1.32	34
2	46	229	226	1.37	58
3	51	231	222	3.87	117
4	51	244	240	1.52	13
5	55	283	275	2.95	47
6	59	375	367	1.93	143
7	64	396	388	1.98	13
8	67	424	409	3.74	16
9	72	446	421	5.97	231
10	78	487	475	2.47	39
11	86	400	394	1.49	208
12	88	503	480	4.71	71
Average				2.78	83

Table 4: The average experiment results for the generated 6 problem sizes.

Group number	Size	Gap (%)	CPU time (seconds)
1	40×3	1.32	34
2	80×3	3.87	118
3	120×3	2.61	156
4	50×4	1.57	29
5	100×4	2.17	126
6	150×4	3.54	265
Average		2.51	121

acceptable computing times (no more than 5 min) by these algorithms. In those experiments, the IP model can provide the cable assignment schemes to the GA as well as the lower bound to evaluate the solution quality. Moreover, we have the following observations:

(1) In Table 2, the proposed hybrid IP/GA approach performs much better than the hybrid MILP/CP approach in solution efficiency particularly to the instances with larger sizes. For the solution quality, they reached the same objective function value for all the instances but the former failed to reduce the gap to 0, meaning that the solution optimality cannot be validated by its own lower bound.

(2) In Tables 3 and 4, the computation times increase with the problem size but the solution quality is relatively stable. It implies that the solution quality is seldom affected by the problem size but depends mainly on the permitted number of iterations, since most of iterations reach up to the limit.

(3) The IP model contributes to improving the solution as well as providing the lower bound. The cable assignment schemes obtained through resolving the IP model, aiming at the optimal objective, can lead the GA to search around the optimum point.

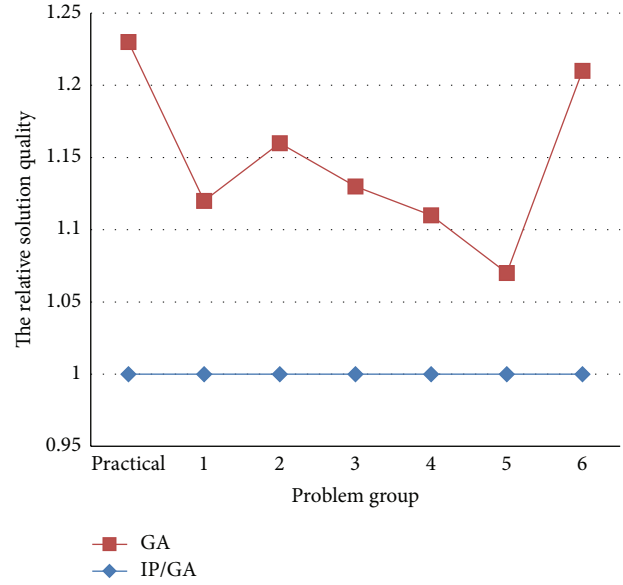


Figure 6: The ARD comparison of the IP/GA from an existing GA.

6. Conclusion

In this paper, a special parallel production lines scheduling problem was studied. A mixed integer programming model considering time window and technical constraints was first formulated. A few valid inequalities were deduced and a hybrid MILP/CP approach was introduced. Based on them, a hybrid IP/GA algorithm was proposed to solve the problem. The numerical experiments were carried out which demonstrated that the proposed solution approach was effective and efficient.

This study highlighted the advantage of combining the traditional exact optimization (IP) with intelligent optimization (GA) method. Other intelligent optimization methods can also be able to hybridize to the IP (or MILP) method. It is more interesting and challenging to hybridize the IP (or MILP) to more elements of the GA (or other heuristics) and deepen the hybridization for the purpose of improving the solution approach.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This research is partly supported by National Natural Science Foundation of China (71201105), China Postdoctoral Science Foundation (2013M530947), and Liaoning Province Universities Scientific Research Project (L2014480).

References

[1] S. Niroomand and B. Vizvari, "Exact mathematical formulations and metaheuristic algorithms for production cost minimization: a case study of the cable industry," *International*

Transactions in Operational Research, vol. 22, no. 3, pp. 519–544, 2015.

[2] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[3] R. M. Karp, "Reducibility among conbinatorial problems," in *Proceedings of the Symposium on the Complexity of Computer Computations*, New York, NY, USA, March 1972.

[4] J. Bruno, J. Coffman, and R. Sethi, "Scheduling independent tasks to reduce mean finishing time," *Communications of the ACM*, vol. 17, pp. 382–387, 1974.

[5] J. K. Lenstra, *Sequencing by Enumerative Methods*, Matematisch Centrum, Amsterdam, Netherlands, 1977.

[6] M. R. Garey and D. S. Johnson, "'Strong' NP-completeness results: motivation, examples, and implications," *Journal of the Association for Computing Machinery*, vol. 25, no. 3, pp. 499–508, 1978.

[7] M. I. Dessouky, B. J. Lageweg, J. K. Lenstra, and S. L. van de Velde, "Scheduling identical jobs on uniform parallel machines," *Statistica Neerlandica*, vol. 44, no. 3, pp. 115–123, 1990.

[8] R. Hübscher and F. Glover, "Applying tabu search with influential diversification to multiprocessor scheduling," *Computers and Operations Research*, vol. 21, no. 8, pp. 877–884, 1994.

[9] B. Srivastava, "An effective heuristic for minimizing makespan on unrelated parallel machines," *Journal of the Operational Research Society*, vol. 49, no. 8, pp. 886–894, 1998.

[10] C. Smutnicki, "On-time scheduling jobs on parallel machines," in *Proceedings of the 16th European Conference on Operational Research*, Brussels, Belgium, 1998.

[11] T. Kämpke, "Simulated annealing: use of a new tool in bin packing," *Annals of Operations Research*, vol. 16, no. 1, pp. 327–332, 1988.

[12] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, 1994.

[13] F. Sivrikaya-Serifoglu and G. Ulusoy, "Parallel machine scheduling with earliness and tardiness penalties," *Computers and Operations Research*, vol. 26, no. 8, pp. 773–787, 1999.

[14] H. I. Stern, *Minimizing Makespan for Independent Jobs on Non-Identical Parallel Machine—An Optimal Procedure*, Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beersheba, Israel, 1976.

[15] H. Belouadah and C. N. Potts, "Scheduling identical parallel machines to minimize total weighted completion time," *Discrete Applied Mathematics*, vol. 48, no. 3, pp. 201–218, 1994.

[16] M. Dell'Amico and S. Martello, "Optimal scheduling of tasks on identical parallel processors," *ORSA Journal on Computing*, vol. 7, no. 2, pp. 191–200, 1995.

[17] S. Martello, F. Soumis, and P. Toth, "Exact and approximation algorithms for makespan minimization on unrelated parallel machines," *Discrete Applied Mathematics*, vol. 75, no. 2, pp. 169–188, 1997.

[18] M. H. Rothkopf, "Scheduling independent tasks on parallel processors," *Management Science*, vol. 12, no. 5, pp. 437–447, 1966.

[19] E. L. Lawler and J. M. Moore, "A functional equation and its application to resource allocation and sequencing problems," *Mgmt Science*, vol. 16, no. 1, pp. 77–84, 1969.

[20] S. K. Sahni, "Algorithms for scheduling independent tasks," *Journal of the Association for Computing Machinery*, vol. 23, no. 1, pp. 116–127, 1976.

[21] Z.-L. Chen and W. B. Powell, "Solving parallel machine scheduling problems by column generation," *INFORMS Journal on Computing*, vol. 11, no. 1, pp. 78–94, 1999.

[22] J. M. van den Akker, J. A. Hoogeveen, and S. L. van de Velde, "Parallel machine scheduling by column generation," *Operations Research*, vol. 47, no. 6, pp. 862–872, 1999.

[23] Z.-L. Chen and W. B. Powell, "A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem," *European Journal of Operational Research*, vol. 116, no. 1, pp. 220–232, 1999.

[24] I. Harjunkoski, V. Jain, and I. E. Grossmann, "Hybrid mixed-integer/constraint logic programming strategies for solving scheduling and combinatorial optimization problems," *Computers & Chemical Engineering*, vol. 24, no. 2–7, pp. 337–343, 2000.

[25] V. Jain and I. E. Grossmann, "Algorithms for hybrid MILP/CP models for a class of optimization problems," *INFORMS Journal on Computing*, vol. 13, no. 4, pp. 258–276, 2001.

[26] J. N. Hooker, "Logic, optimization, and constraint programming," *INFORMS Journal on Computing*, vol. 14, no. 4, pp. 295–321, 2002.

[27] C. T. Maravelias and I. E. Grossmann, "A hybrid MILP/CP decomposition approach for the continuous time scheduling of multipurpose batch plants," *Computers & Chemical Engineering*, vol. 28, no. 10, pp. 1921–1949, 2004.

[28] H. Ren and S. Liu, "Study on the hybrid IP/CP approach for a multi-machine assignment scheduling problem," in *Proceedings of the International Conference on Management and Engineering (CME '14)*, Shanghai, China, 2014.

[29] E. Horowitz and S. Sahni, "Exact and approximate algorithms for scheduling non-identical processors," *Journal of the Association for Computing Machinery*, vol. 23, no. 2, pp. 317–327, 1976.

[30] D. E. Goldberg, *Genetic Algorithms*, Addison-Wesley, Reading, Mass, USA, 1989.

[31] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.

[32] P. S. Oliveto and C. Witt, "On the runtime analysis of the simple genetic algorithm," *Theoretical Computer Science*, vol. 545, no. 6, pp. 2–19, 2014.

[33] S. M. Lee and A. A. Asllani, "Job scheduling with dual criteria and sequence-dependent setups: mathematical versus genetic programming," *OMEGA, The International Journal of Management Science*, vol. 32, no. 2, pp. 145–153, 2004.

[34] R. Ruiz, C. Maroto, and J. Alcaraz, "Two new robust genetic algorithms for the flowshop scheduling problem," *Omega*, vol. 34, no. 5, pp. 461–476, 2006.

[35] M. Kapanoglu and I. O. Koc, "A multi-population parallel genetic algorithm for highly constrained continuous galvanizing line scheduling," in *Hybrid Metaheuristics*, vol. 4030 of *Lecture Note in Computer Science*, pp. 28–41, Springer, Berlin, Germany, 2006.

[36] C. H. Martin, "A hybrid genetic algorithm/mathematical programming approach to the multi-family flowshop scheduling problem with lot streaming," *Omega*, vol. 37, no. 1, pp. 126–137, 2009.

[37] J.-W. Chung, S.-M. Oh, and I.-C. Choi, "A hybrid genetic algorithm for train sequencing in the Korean railway," *Omega*, vol. 37, no. 3, pp. 555–565, 2009.

[38] W. Yin, M. Liu, and C. Wu, "A new genetic algorithm for parallel machine scheduling with process constraint," *Acta Electronica Sinica*, vol. 29, no. 11, pp. 1482–1485, 2001.

[39] J. X. Luo and L. X. Tang, "A new ILS & SS algorithm for parallel-machine scheduling problem," *Acta Electronica Sinica*, vol. 31, no. 6, pp. 917–924, 2005.

[40] E. Vallada and R. Ruiz, "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times," *European Journal of Operational Research*, vol. 211, no. 3, pp. 612–622, 2011.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2016, Article ID 3512546, 13 pages
http://dx.doi.org/10.1155/2016/3512546

Research Article

A New Adaptive Hungarian Mating Scheme in Genetic Algorithms

Chanju Jung,[1] Yong-Hyuk Kim,[2] Yourim Yoon,[3] and Byung-Ro Moon[1]

[1]*Department of Computer Science and Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea*
[2]*Department of Computer Science and Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01890, Republic of Korea*
[3]*Department of Computer Engineering, Gachon University, 1342 Sengnamdaero, Sujeong-gu, Seongnam-si, Gyeonggi-do 13120, Republic of Korea*

Correspondence should be addressed to Byung-Ro Moon; moon@snu.ac.kr

Received 10 December 2015; Revised 1 March 2016; Accepted 17 March 2016

Academic Editor: Lu Zhen

In genetic algorithms, selection or mating scheme is one of the important operations. In this paper, we suggest an adaptive mating scheme using previously suggested Hungarian mating schemes. Hungarian mating schemes consist of maximizing the sum of mating distances, minimizing the sum, and random matching. We propose an algorithm to elect one of these Hungarian mating schemes. Every mated pair of solutions has to vote for the next generation mating scheme. The distance between parents and the distance between parent and offspring are considered when they vote. Well-known combinatorial optimization problems, the traveling salesperson problem, and the graph bisection problem are used for the test bed of our method. Our adaptive strategy showed better results than not only pure and previous hybrid schemes but also existing distance-based mating schemes.

1. Introduction

Mating scheme or selection is one of the important operations in genetic algorithm (GA). Most operations in GAs are closely related to the performance. These operations interact with each other. A small change of a key operation may cause a dramatic change in result. Ochoa et al. [1] presented that assortative mating is a good choice when the mutation rate is high, while disassortative mating is a good choice when the mutation rate is low.

In mating or selection stage, the methods of mating are classified into three groups. The first one gives preference to similar solutions [2, 3]. This method focuses on exploitation. It assumes that similar solutions have higher chance to make better solutions because they use discovered good schemata.

The second one is dissimilar mating. This method focuses on exploration, and it tries to evade a premature convergence and a fast diversity consumption of similar mating. It is realized mostly by a restriction. Ramezani and Lotfi [4] restricted mating between family solutions such as parent and offspring. They solved function optimization problems and obtained good results. Fernandes et al. [5] reported that dissimilar mating outperforms a simple GA or similar mating in vector quantization problem.

The last group tries to find a better mating scheme by combining two or more mating schemes. Ishibuchi et al. [6, 7] considered the number of mating candidates. They used their parameters to control similarity of a mated solution. They proposed a method of changing its controlling parameter at a middle of running. They also presented that the changing parameter made better results (nearer to Pareto-optimal solutions) than fixed parameters.

Galán et al. [8] proposed a mating scheme where each individual has its mating preference value to balance exploitation and exploration. A low value of mating preference

makes a match between solutions close to each other, while a high mating preference makes a match between solutions far from each other. The preference is inherited or mutated like a normal gene. They tested their scheme in various environments of function optimization. They showed that their scheme outperformed random mating or a scheme with a fixed preference value.

In this paper, we propose a new adaptive Hungarian mating scheme. Our hybrid scheme adaptively selects a pure scheme for the next generation with voting. The traveling salesperson problem (TSP) and the graph bisection problem are the test problems of our adaptive scheme. We summarize our contributions as follows. (i) We propose a new adaptive mating scheme, (ii) we provide a reasonable explanation for our scheme, (iii) we show that our adaptive hybrid scheme is more effective than any pure Hungarian scheme and a previous simple hybrid scheme for the two test problems, and (iv) finally we show that our scheme changes its action as the size of problem space changes.

In comparison with the preliminary version of this paper [9], we provide the following extended information: (i) empirical observations and theoretical proof of the recommended parameter settings, (ii) detailed results to show the search behaviors of our algorithm, (iii) comparison of solution qualities with existing distance-based mating methods, and (iv) analysis on computation time of the proposed method and existing ones.

The remainder of this paper is divided into five parts. Previously suggested Hungarian mating schemes and their hybrid strategy are presented in Section 2. Motivation of this study is described in Section 3. Section 4 explains our adaptive hybrid mating scheme. In Section 5, we give experimental results and analyze the results. In Section 6, we draw conclusions.

2. Background

2.1. Hungarian Method. Assume a weighted complete bipartite graph with bipartition (X, Y): $X = x_1, x_2, \ldots, x_{n/2}$, $Y = y_1, y_2, \ldots, y_{n/2}$, and each edge $(x_i, y_j) \in X \times Y$ has its weight w_{ij}. For optimal matching, we consider a problem of finding a maximum (or minimum) weight bijective matching as follows:

$$\max_{\sigma \in \sum_{n/2}} \left(\sum_{i=1}^{n/2} w_{i\sigma(i)} \right) \text{ or } \min_{\sigma \in \sum_{n/2}} \left(\sum_{i=1}^{n/2} w_{i\sigma(i)} \right), \quad (1)$$

where σ is a permutation of size $n/2$.

Optimum matching can be computed by the Hungarian method [14]. It can be computed in $O(n^3)$ time [15]. Avis [16] has suggested an approximation algorithm of $O(n^2)$ time. The Hungarian method has been utilized in various studies [17–20].

2.2. Hungarian Mating Scheme. Recently, we suggested Hungarian mating schemes [10]. We divided population into the same number of female and male individuals. The Hungarian method is applied to mating. In each problem, a suited distance metric is adopted. It will be described in the next subsection.

Figure 1 describes the mating schemes. Distribution of an example population is illustrated in Figure 1(a). Thirty individuals are shown: fifteen plus symbols (+) are female solutions, and fifteen filled circles (•) are male ones. The Euclidean distance is adopted as the distance metric. Figure 1(b) shows a result of random mating scheme which is called "RAND." The result of minimizing the sum of distances is displayed in Figure 1(c). The result is obtained from the following formula with the Hungarian method:

$$\min_{\sigma \in \sum_{n/2}} \sum_{i=1}^{n/2} d\left(m_i, f_{\sigma(i)}\right), \quad (2)$$

where m_i's are male solutions, $f_{\sigma(i)}$'s are their mated female ones, and d is the Euclidean distance function. The scheme in Figure 1(c) will be called "NEAR" in this study. Figure 1(d) shows the result of maximizing the sum of distances. The result is obtained from the following formula with the Hungarian method:

$$\max_{\sigma \in \sum_{n/2}} \sum_{i=1}^{n/2} d\left(m_i, f_{\sigma(i)}\right). \quad (3)$$

This scheme is called "FAR."

NEAR method extremely concentrates on exploitation, while FAR method extremely focuses on exploration. In bijective mating, NEAR method minimizes the sum of distances. It is an extreme strategy to decrease diversity and use inherent parts of solutions. Similarly, FAR method is an extreme method to preserve diversity.

2.3. Test Problems and Distance Metric. Our test problems are TSP and the graph bisection problem. In TSP, a complete undirected graph G is given. Each edge in G has a nonnegative weight. The objective of the problem is to find a minimum tour that passes through all the vertices (i.e., Hamiltonian cycle) of G.

Assume an undirected graph $G = (V, E)$: V is a vertex set and E is an edge set. K-way partitioning is defined as partitioning the vertex set V into K disjoint subsets $C_1, C_2, \ldots, C_K$. A K-way partition is said to be balanced if the difference of cardinalities between the largest and the smallest subsets is zero or one. The cut size of a partition is the number of edges with endpoints in different subsets of the partition. The K-way partitioning problem is the problem of finding a K-way balanced partition with minimum cut size. In this paper, we set K to be equal to two and call this problem "graph bisection."

The distance metric is the same as that used in our previous study [10]. In TSP, for the phenotype distance metric, the quotient swap distance [20] was used. The quotient swap distance is defined as the smallest one among swap distances. The swap distance between X and Y is the minimum number of swaps to make X be equal to Y. In other words, between X and every shifted Y the minimum value of the swap distance is the quotient swap distance between X and Y.

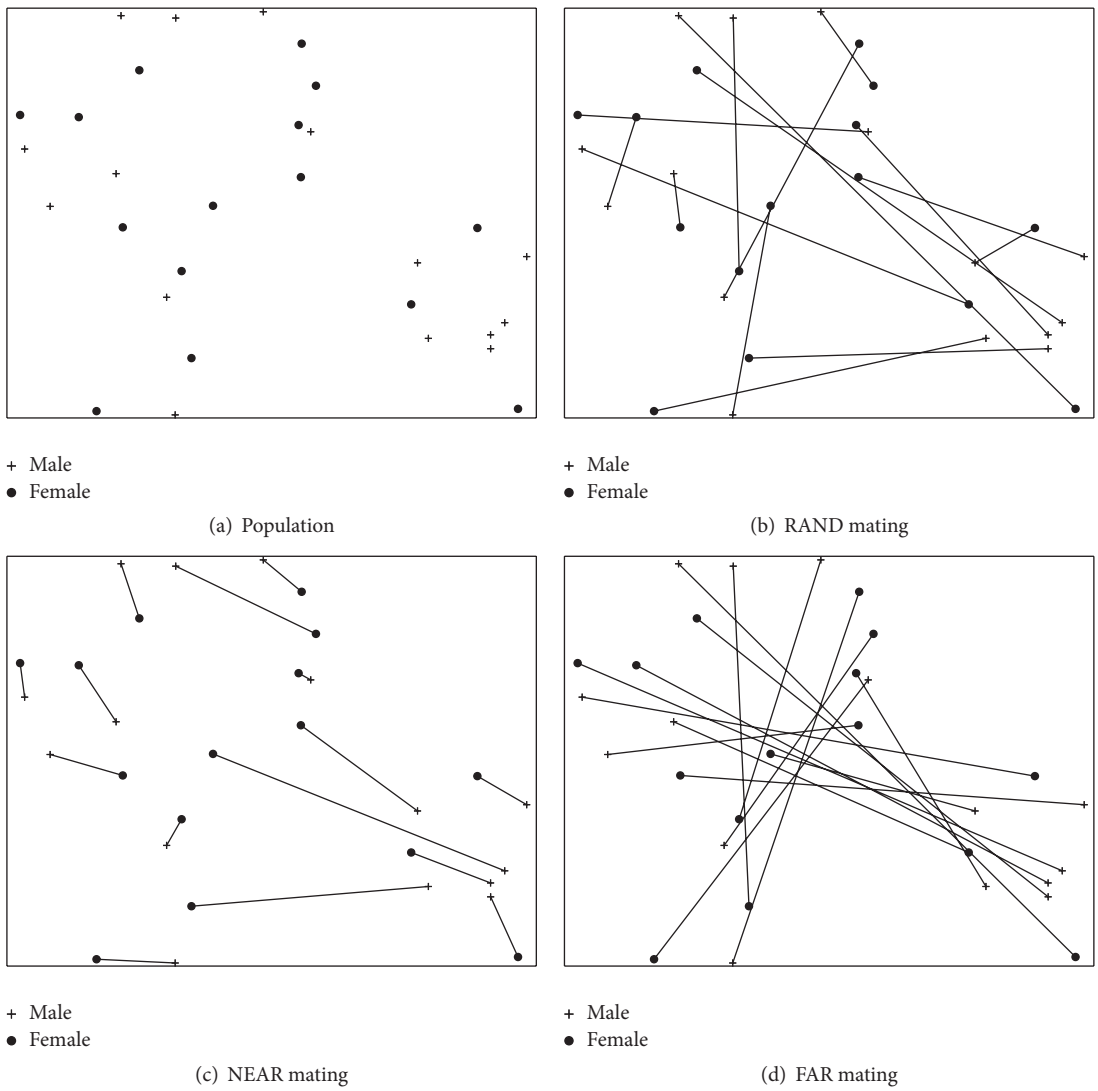


Figure 1: Hungarian mating schemes [10].

In graph bisection, for the phenotype distance metric, the quotient Hamming distance [20] was used. Similar to the quotient swap distance, the quotient Hamming distance is defined as the smallest one among Hamming distances. The Hamming distance between two strings is defined by the number of different positions at which the corresponding symbols are different. In other words, the quotient Hamming distances between X and Y are the smaller value of the Hamming distance between X and Y and that between X and $\overline{Y}$.

3. Motivation

We reported that the best Hungarian mating scheme varies according to problems and their sizes [10]. We proposed a simple hybrid scheme of changing the mating scheme from NEAR to RAND at the mth generation in TSP, where m is the number of cities. The hybrid scheme for graph bisection changes mating scheme from RAND to FAR at the 100th generation. Parameters m and 100 are based on some empirical observation. But the hybrid scheme also has a weakness. The switching time before running of GAs was predetermined. So it is hard to apply the method to new problems or instances.

Galán et al. [8] reported that a self-adaptive mating scheme can be better than traditional random mating and their best-first mating and best-last mating. In the best-first mating, each solution pairs up with its nearest one in the order from the best solution to the worst one. In contrast, in the best-last mating, each solution pairs up with its farthest one in the order from the best solution to the worst one. The best-first mating resembles NEAR method as the best-last mating resembles FAR method. NEAR and FAR are extreme cases of mating. The ideal mating scheme may exist in some middle point of NEAR and FAR as Galán et al. [8] showed in function optimization.

Suggesting a new adaptive hybrid scheme of the Hungarian mating schemes is our main goal. A good scheme may (i) work irrespective of problems or instances, (ii) select its action as environment changes, and (iii) show better results than consistently applied pure scheme. A new scheme with these features will be proposed.

In this paper, we show the influence of the proposed mating scheme, not aiming to beat the state-of-the-art results of TSP and graph bisection. Local optimization may play an important role in making practical solutions. We do not use any local optimization. Instead, we concentrate on balance between exploration and exploitation with a new mating scheme.

4. Proposed Method

4.1. Voting Rules. We assume the same number of male and female solutions as Goh et al. did in [21]. In each generation, our method selects FAR, RAND, or NEAR for the next generation. Our method does not simulate three schemes as they are. Instead, the appropriate scheme is adaptively adopted. For that, a mating scheme for the next generation is selected with majority voting. Every matched pair has to vote. Our crossover operator generates two offspring, and their gender is randomly assigned. The voting is carried out after mutation. So our voting algorithm compares two parents and two offspring after mutation. The rules of voting are described by Algorithm 1. If one of the parents is the same as its son or daughter, this pair votes for FAR scheme. Otherwise, a ratio of distance between parents over the sum of the mother-daughter distance and the father-son distance is considered. In the case where the ratio is less than α, this pair votes for FAR scheme. If the ratio is equal to or greater than α and less than β, this pair votes for RAND scheme. The remaining case is where the ratio is equal to or greater than β. In this case, this pair votes for NEAR scheme. In the next generation, the scheme which gets the most votes is adopted.

```
// input: two parents and two offspring
// output: FAR, NEAR, or RAND
// d(x, y): distance function between x and y
Function vote(p1, p2, o1, o2)
{
    If d(p1, p2) = 0, d(p1, o1) = 0, or d(o2, p2) = 0 then
      return FAR;
    end if
    ratio ← d(p1, p2)/(d(p1, o1) + d(o2, p2));
    if ratio < α then
      return FAR;
    end if
    if α ≤ ratio < β then
      return RAND;
    end if
    if ratio ≥ β then
      return NEAR;
    end if
}
```

Algorithm 1: Voting rules.

4.2. Parameter Setting. We set α to be 0.5 and β to be 1. Figure 2 describes the median of the ratio values according to generation for an instance of each test problem. x-axis represents generation and y-axis represents the ratio values.

We call the median of the ratio values after crossover (before mutation) (thin line) BM. The median of the ratio values after mutation (thick points) is called AM. After crossover (before mutation), most of BM values are close to 1. BM does not change much, while the diversity of population decreases. On the other hand, AM drops slowly as the diversity decreases.

A mutation operator moves an individual to nearby space. The distribution of moving distance by a mutation is independent of the distance between parents. The expected value of BM is one when we use a geometric crossover [22]. It will be proven in the next subsection. AM values over 1 appear frequently when the distance between parents is long enough. It means that we have sufficient diversity to consume. So a family votes for NEAR. Besides, the lower bound of BM is 0.5 when we use a geometric crossover. It will also be proven in the next subsection. AM values below 0.5 appear due to mutation effect. They are observed when the distance between parents is very close to 0. So a family votes for FAR. In other words, an influence of the mutation is estimated by the distance between parents. High influence of the mutation or a low AM value means that the matched parents are too close to each other to produce new solutions, while low influence of the mutation or a high AM value means that the parents are far from each other so we can match nearer solutions.

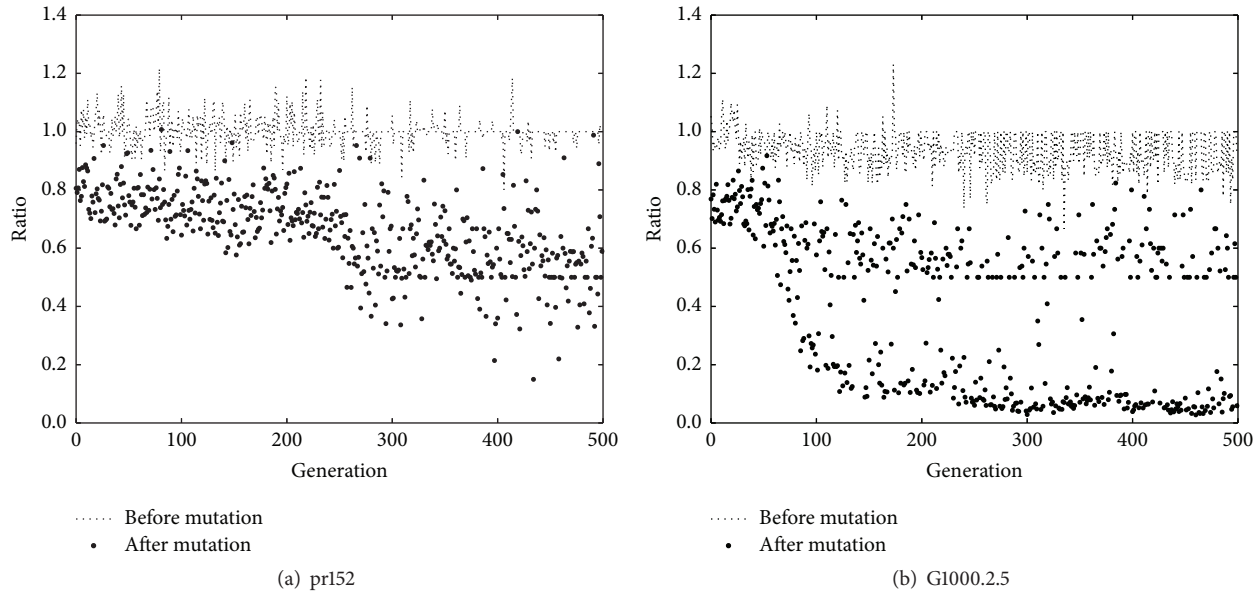
4.3. Theoretical Support. A binary crossover operator is geometric if all offspring are in a convex segment between parents. That is, $d(p_1, p_2) = d(p_1, o) + d(o, p_2)$, where $d(p_1, p_2)$ is a distance between p_1 and p_2, p_i's are parents, and o is an offspring obtained from a geometric crossover. Let D be the distance between both parents. We assume that $D = d(p_1, p_2) \neq 0$, crossover is geometric [22], $p_1 \neq p_2$, $p_1 \neq o_1$, and $p_2 \neq o_2$. We remind the reader that our ratio value is defined as

$$\frac{d(p_1, p_2)}{d(p_1, o_1) + d(o_2, p_2)}, \tag{4}$$

where o_1 and o_2 are offspring obtained from a geometric crossover between p_1 and p_2.

Proposition 1. *Under these assumptions, the expected value of one's ratio is 1. That is,*

$$E\left[\frac{d(p_1, p_2)}{d(p_1, o_1) + d(o_2, p_2)}\right] = 1. \tag{5}$$



(a) pr152 (b) G1000.2.5

Figure 2: Median of ratio values according to generation.

Proof. It is enough to show that

$$\begin{aligned}
&E\left[d\left(p_1,o_1\right)+d\left(o_2,p_2\right)\right]=D,\\
&E\left[d\left(p_1,o_1\right)+d\left(o_2,p_2\right)\right]\\
&\quad=E\left[d\left(p_1,o_1\right)\right]+E\left[d\left(o_2,p_2\right)\right]\\
&\qquad\qquad\left(\because E\left[\cdot\right]\text{ is linear}\right)\\
&\quad=E\left[d\left(p_1,o_1\right)\right]+E\left[d\left(p_1,p_2\right)-d\left(p_1,o_2\right)\right]\\
&\qquad\qquad\left(\because \text{Crossover is geometric}\right)\\
&\quad=E\left[d\left(p_1,o_1\right)\right]+E\left[D-d\left(p_1,o_2\right)\right]\\
&\quad=E\left[d\left(p_1,o_1\right)\right]+D-E\left[d\left(p_1,o_2\right)\right]\\
&\qquad\qquad\left(\because E\left[\cdot\right]\text{ is linear}\right)\\
&\quad=D\quad\left(\because E\left[d\left(p_1,o_1\right)\right]=E\left[d\left(p_1,o_2\right)\right]\right).
\end{aligned}\tag{6}$$

□

Proposition 2. *Under the same assumptions, the lower bound of one's ratio value is 0.5. That is,*

$$\frac{d\left(p_1,p_2\right)}{d\left(p_1,o_1\right)+d\left(o_2,p_2\right)}\geq\frac{1}{2}.\tag{7}$$

Proof. By the assumption of geometric crossover,

$$\begin{aligned}
d\left(p_1,p_2\right)&\geq d\left(p_1,o_1\right),\\
d\left(p_1,p_2\right)&\geq d\left(o_2,p_2\right).
\end{aligned}\tag{8}$$

By summing the above inequalities, $2d(p_1,p_2)\geq d(p_1,o_1)+d(o_2,p_2)$. Hence, we obtain

$$\frac{d\left(p_1,p_2\right)}{d\left(p_1,o_1\right)+d\left(o_2,p_2\right)}\geq\frac{1}{2}.\tag{9}$$

□

5. Experiments

5.1. Tested GA. We use a generational GA for test. Each solution is bijectively matched with opposite gender. A couple of individuals produce one female solution and one male solution. The genders are randomly assigned. We used 50 female and male individuals. We applied elitism [23] as a replacement strategy in both genders. Fifty best solutions remained for the next generation among previous 50 parents and new 50 offspring. Table 1 gives the other genetic operator settings.

5.2. Traveling Salesman Problem. From TSPLIB [24], four Euclidean instances are selected: berlin52, kroA100, bier127, and pr152. In each instance, the number of cities is represented in the right part of the name.

Figure 3 displays the fitness of the best individual according to generation. The average of the best fitness values (Avg) and the standard deviation (Std) per 200 generations are shown in Table 2. Method "single best" denotes the best single result among RAND, NEAR, and FAR in each generation. Method "simple hybrid" is the strategy that changes mating scheme once to a proper one. It was introduced in our previous work [10]. The results of all figures and tables are the average values over 1,000 runs.

Table 1: Genetic parameter settings.

	TSP	Graph bisection problem
Representation	Order-based encoding	Assignment of one gene for each vertex (zero or one)
Recombination	Partially matched crossover [11]	One-point crossover
Mutation	Double-bridge kick move [12] (50%)	Random swap of some pairs of genes (50%)
Repair	—	Random repair until partition is feasible
Stopping criterion	1,000 generations	500 generations

Table 2: Results of TSP.

Problem instance	Method	Gen 200 Avg (Std)	Gen 400 Avg (Std)	Gen 600 Avg (Std)	Gen 800 Avg (Std)	Final (= 1000) Avg (Std)
berlin52	Single best	1.11*e*4 (5.59*e*2)	9.71*e*3 (4.23*e*2)	9.34*e*3 (3.26*e*2)	9.71*e*3 (2.85*e*2)	9.07*e*3 (3.08*e*2)
	Simple hybrid	1.08*e*4 (4.34*e*2)	9.72*e*3 (3.75*e*2)	9.39*e*3 (3.48*e*2)	9.22*e*3 (3.37*e*2)	9.10*e*3 (3.18*e*2)
	New method	**1.07*e*4 (4.68*e*2)**	**9.63*e*3 (3.69*e*2)**	**9.30*e*4 (3.45*e*2)**	**9.13*e*3 (3.23*e*2)**	**9.02*e*3 (3.09*e*2)**
kroA100	Single best	6.87*e*4 (4.10*e*3)	4.84*e*4 (2.50*e*3)	4.13*e*4 (1.71*e*3)	3.85*e*4 (1.80*e*3)	3.65*e*4 (9.03*e*2)
	Simple hybrid	**6.62*e*4 (3.20*e*3)**	**4.76*e*4 (2.22*e*3)**	4.10*e*4 (1.74*e*3)	3.78*e*4 (1.58*e*3)	3.60*e*4 (1.41*e*3)
	New method	6.78*e*4 (4.09*e*3)	4.79*e*4 (2.40*e*3)	**4.04*e*4 (1.76*e*3)**	**3.71*e*4 (1.56*e*3)**	**3.51*e*4 (1.48*e*3)**
bier127	Single best	3.36*e*5 (9.09*e*3)	2.52*e*5 (9.44*e*3)	2.18*e*5 (7.03*e*3)	2.00*e*5 (6.76*e*3)	1.90*e*5 (2.25*e*3)
	Simple hybrid	**3.27*e*5 (1.09*e*4)**	**2.49*e*5 (9.06*e*3)**	**2.16*e*5 (7.22*e*3)**	1.99*e*5 (6.70*e*3)	1.89*e*5 (6.53*e*3)
	New method	3.38*e*5 (1.38*e*4)	2.52*e*5 (9.79*e*3)	2.57*e*5 (7.83*e*3)	**1.97*e*5 (6.80*e*3)**	**1.87*e*5 (6.20*e*3)**
pr152	Single best	5.07*e*5 (1.66*e*4)	3.28*e*5 (3.90*e*4)	2.47*e*5 (1.36*e*4)	2.05*e*5 (1.13*e*4)	1.81*e*5 (8.36*e*3)
	Simple hybrid	**4.78*e*5 (1.75*e*4)**	**3.14*e*5 (1.59*e*4)**	**2.40*e*5 (1.32*e*4)**	2.01*e*5 (1.12*e*4)	1.79*e*5 (9.39*e*3)
	New method	5.09*e*5 (2.65*e*4)	3.30*e*5 (2.21*e*4)	2.45*e*5 (1.60*e*4)	**2.01*e*5 (1.23*e*4)**	**1.76*e*5 (1.00*e*4)**

CPU: Intel Xeon E5530 2.40 GHz. Average from 1,000 runs.
Avg: average (the smaller, the better); Std: standard deviation.

Table 3: Statistical test of TSP.

Problem instance	Compared method	t-test	p value
berlin52	Single best	+	$1.33e-03$
	Simple hybrid	+	$5.71e-09$
kroA100	Single best	+	$1.0e-103$
	Simple hybrid	+	$4.01e-42$
bier127	Single best	+	$2.41e-42$
	Simple hybrid	+	$9.72e-18$
pr152	Single best	+	$4.46e-26$
	Simple hybrid	+	$2.01e-10$

p value: the smaller, the more significant.
+: significantly better under level $1.00e-02$.

In early stages of each run, the simple hybrid method showed the best fitness. But at the end of each run, our method outperformed the others for all instances.

Table 3 shows the statistical test result of significance for Table 2. We used Welch's t-test [25]. t-value of $A-B$ in Table 3 is computed as follows:

$$t = \frac{\overline{X_A} - \overline{X_B}}{\sqrt{S_A^2/n_A + S_B^2/n_B}}, \tag{10}$$

where $\overline{X_A}$ is the average of A, S_A is the standard deviation of A, and n_A is the test number of A. The more significant result causes the lower p value. In most cases, p values are very close to zero. A plus mark (+) denotes that our scheme has passed t-test under significance level, 0.01. For all instances, our scheme is significantly better than the others.

Figure 4 shows the average voting rate of three schemes. The graph shows the average over 1,000 runs. In the early stage, RAND and NEAR get higher chance to be elected. NEAR is rarely selected in the early stage. As the diversity decreases, supporters of FAR increase. At the end of each run, almost all families vote for FAR. When we compare four instances in Figure 4, we can conclude that our method is adaptive. Consuming a diversity in a small space is faster than that in a large space. So our algorithm changes the mating scheme from RAND (or rarely NEAR) to FAR. The speed of changing scheme for instance pr152 was slower than that for instance berlin52.

5.3. Graph Bisection Problem. We used four popular instances with 1,000 vertices [26]. The difference of the instances is edge density. The right part of each name (such as 05) represents the average vertex degree.

Figure 5 illustrates the fitness of the best individual over all generations. The results of all figures and tables are written with the average values over 1,000 runs as in TSP. The average of the best fitness values (Avg) and the standard deviation (Std) per 100 generations are listed in

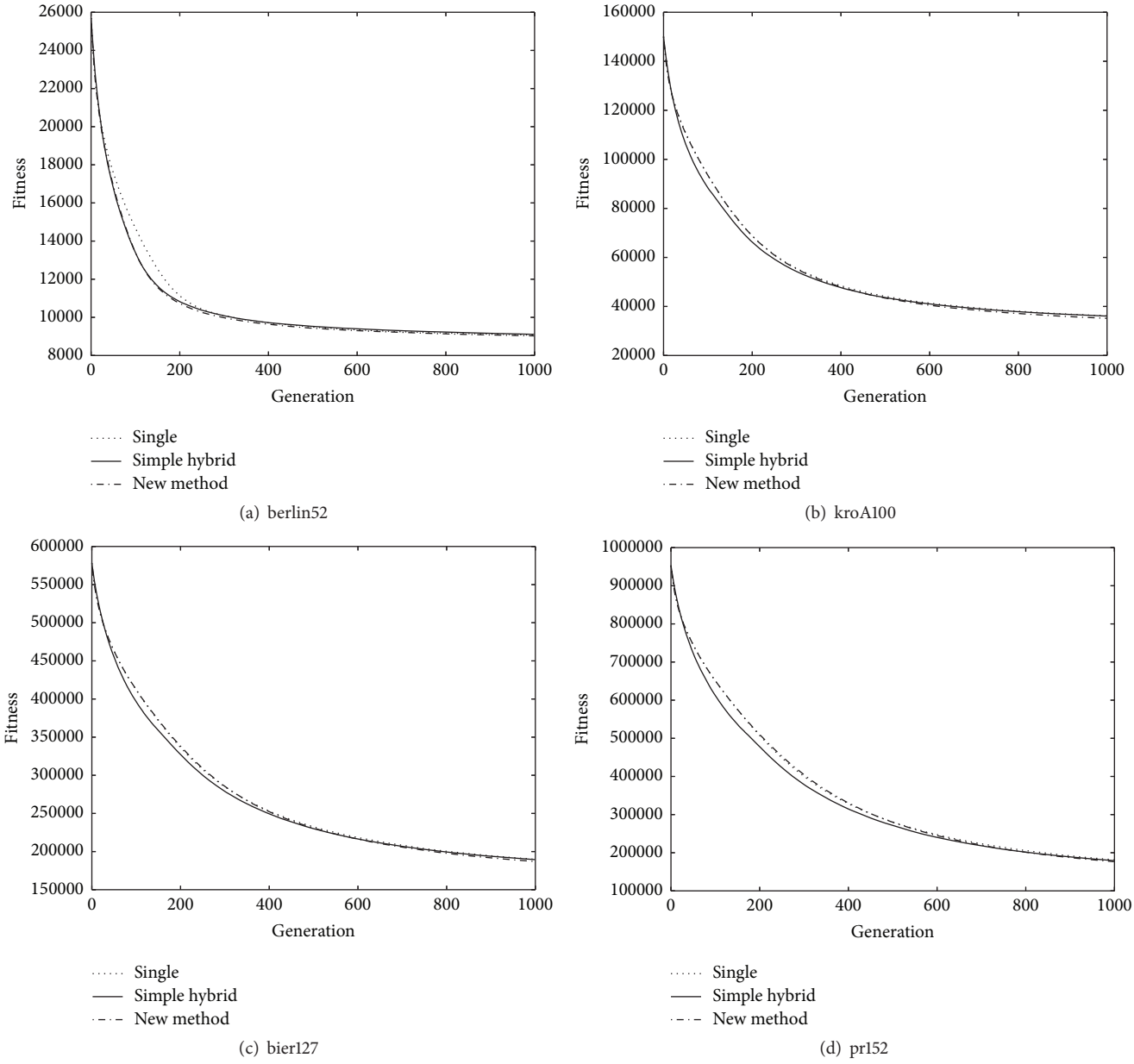


Figure 3: Fitness of mating schemes in TSP (the smaller, the better).

Table 4. Method "single best" and method "simple hybrid" are the same as in TSP. In almost all generations of all the instances, our method outperformed the others. Table 5 shows t-test for the results in Table 4. It is conducted in the same way as that used in TSP. In all instances except one our scheme was significantly superior to the others.

Figure 6 shows the average voting rate of three schemes. While TSP showed different speed of changing schemes according to each instance, the figures of four graph bisection instances are almost the same as each other. Previously we reported that NEAR showed very poor results in this problem [10]. With our new method, NEAR is naturally abandoned because FAR increases very fast.

5.4. Comparison with Existing Methods. We compared our method with existing distance-based mating ones. We implemented variants of Ishibuchi and Shibata's [13] and Galán et al.'s [8] methods with two same-sized genders. Ishibuchi and Shibata's method [13] selects one parent that is the farthest individual from the average among the results of repeated tournament selections of α times. Their method selects the other parent that is the nearest individual from the first parent among the results of repeated tournament selections of β times. We set α and β to be 9 as in [13]. The transformed variant selects the first parent from the female solutions and selects the second parent from the male solutions. It is repeated until all solutions are one-to-one matched. Galán et al.'s method [8] selects one parent that is the best. As the other

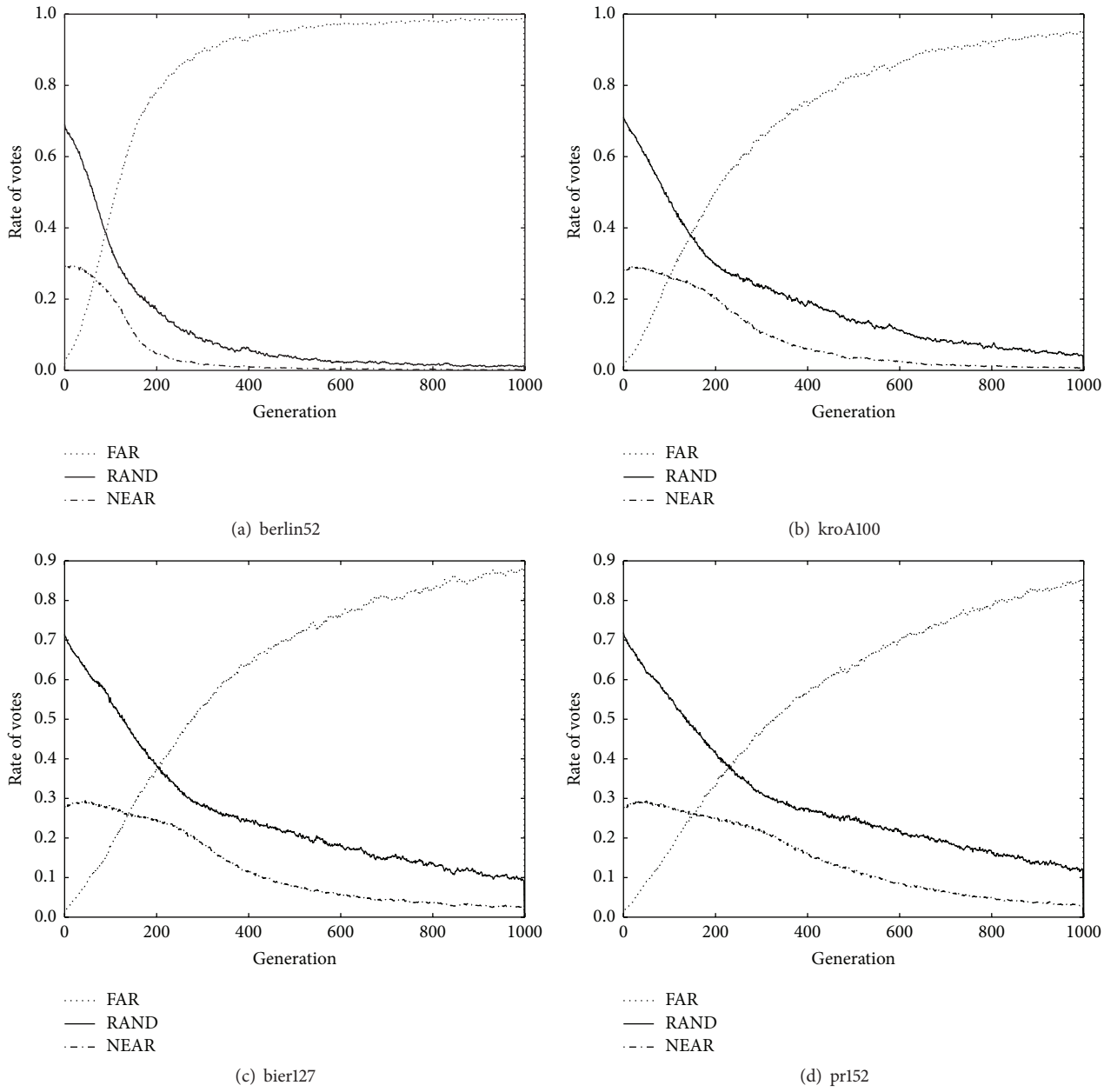


Figure 4: Voting rates of schemes in TSP.

parent, their method selects the $(\gamma - 1)$th nearest individual, where γ is the mating preference of the first parent. The mating preference is inherited in crossover, and it increases by 1 with probability 0.25 or decreases by 1 with probability 0.25, in mutation. The same as the variant of Ishibuchi and Shibata's method [13], we made this method select the first parent from the female solutions and the second parent from the male solutions. It is repeated until all solutions are one-to-one matched. All the conditions and settings excluding mating are the same as those in the experiments of the previous sections.

Table 6 compares the solution qualities of these two existing methods and ours. For all instances of two test problems, our method significantly outperformed the others. Table 7 compares the computation times with respect to mating. Each value in Table 7 except mating proportion is measured in seconds. Our method took more time than Galán's method. But our method was faster than Ishibuchi's. Galán's method repeats finding the $(\gamma - 1)$th nearest individual, whereas our method maximizes (or minimizes) the sum of distances. For graph bisection problem, computation times of instances of our method are similar to each other because the instances have the same number of nodes. In TSP, as the solution space grows, the proportion of mating time decreases, because the mating time of our method is mainly bounded by population size. As distance scale grows, mating time increases. It can be resolved by approximating the scale of distance values.

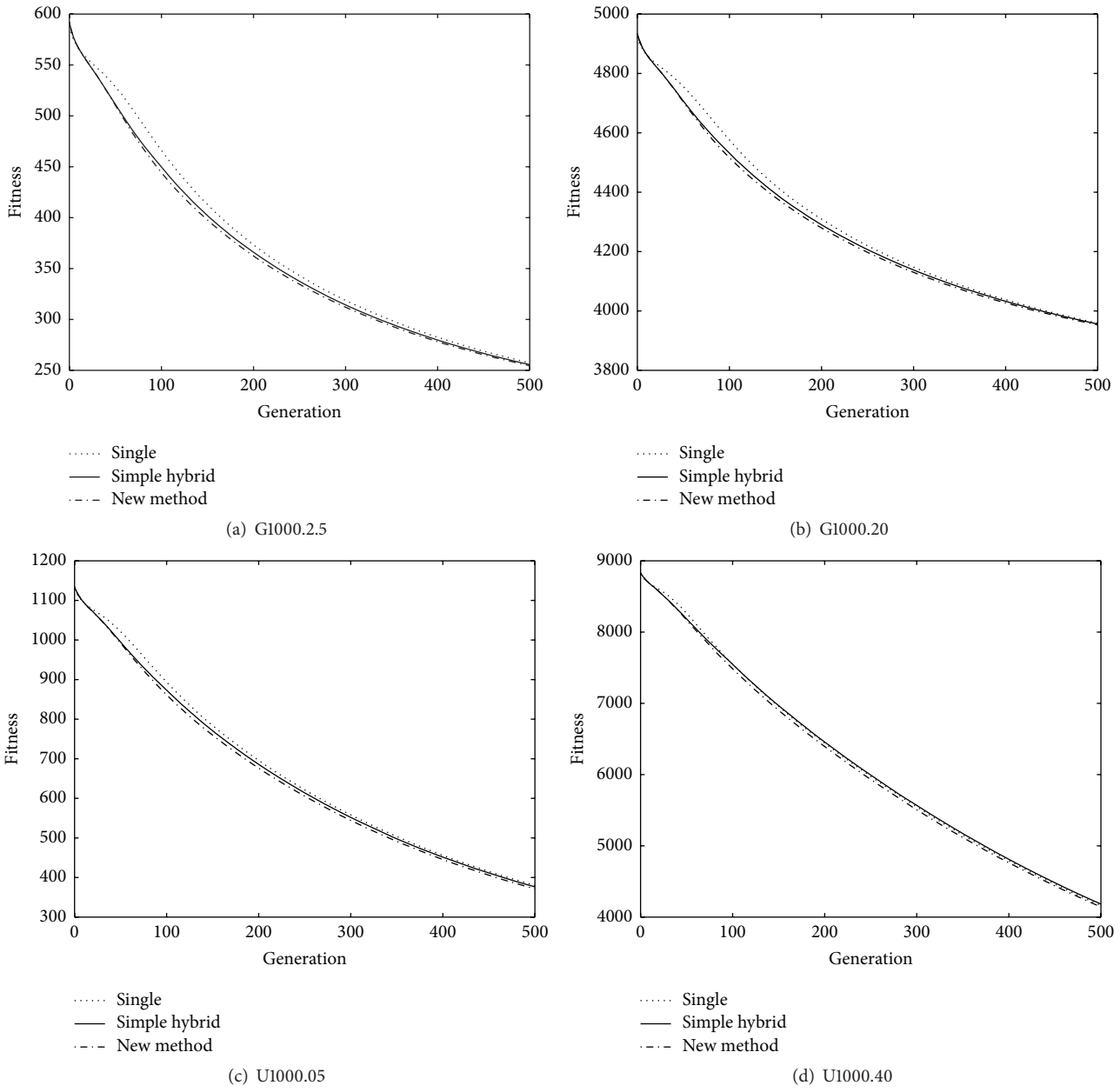


Figure 5: Fitness of mating schemes in graph bisection (the smaller, the better).

Our mating method did not overburden the entire GA, and also we expect reducing time burden through some improved implementation.

6. Concluding Remarks

We analyzed the proposed adaptive hybrid mating scheme for TSP and graph bisection. FAR scheme is biased on exploitation, while NEAR scheme is biased on exploitation. Our mating scheme assesses the distance of the matched parents with their offspring. With this assessment, our adaptive scheme tries to find a balanced point between exploration and exploitation in each generation. We also compared the proposed method with two existing distance-based methods. The proposed method showed better performance than the two existing methods.

We set the threshold parameters as 0.5 and 1.0 with some observation and the values are theoretically justified. But we expect that the method of dynamically adjusting these values may produce better results. Real-coded problems may have different features from combinatorial optimization. With our scheme, more various problems such as function optimization can be tested. There are still opportunities for further enhancements and we will study the presented method

Table 4: Results of graph bisection.

Problem instance	Method	Gen 100 Avg (Std)	Gen 200 Avg (Std)	Gen 300 Avg (Std)	Gen 400 Avg (Std)	Final (= 500) Avg (Std)
G1000.2.5	Single best	4.48*e*2 (1.10*e*1)	3.71*e*2 (9.72*e*0)	3.19*e*2 (1.17*e*1)	2.83*e*2 (1.02*e*1)	2.57*e*2 (9.27*e*0)
	Simple hybrid	4.50*e*2 (1.17*e*1)	3.66*e*2 (9.36*e*0)	3.14*e*2 (9.03*e*0)	2.80*e*2 (8.82*e*0)	2.55*e*2 (1.02*e*1)
	New method	**4.44*e*2 (1.16*e*1)**	**3.62*e*2 (9.55*e*0)**	**3.11*e*2 (9.18*e*0)**	**2.78*e*2 (8.77*e*0)**	**2.54*e*2 (8.68*e*0)**
G1000.20	Single best	4.58*e*3 (3.11*e*1)	4.30*e*3 (2.78*e*1)	4.15*e*3 (2.70*e*1)	4.04*e*3 (3.16*e*1)	3.96*e*3 (3.00*e*1)
	Simple hybrid	4.53*e*3 (3.11*e*1)	4.29*e*3 (2.83*e*1)	4.13*e*3 (2.78*e*1)	4.03*e*3 (2.72*e*1)	3.96*e*3 (2.71*e*1)
	New method	**4.51*e*3 (3.43*e*1)**	**4.28*e*3 (2.90*e*1)**	**4.13*e*3 (2.72*e*1)**	**4.02*e*3 (2.64*e*1)**	**3.95*e*3 (2.61*e*1)**
U1000.05	Single best	8.71*e*2 (2.14*e*1)	6.96*e*2 (3.11*e*1)	5.58*e*2 (2.77*e*1)	4.56*e*2 (2.55*e*1)	3.80*e*2 (2.41*e*1)
	Simple hybrid	8.75*e*2 (2.20*e*1)	6.86*e*2 (2.25*e*1)	5.52*e*2 (2.26*e*1)	4.51*e*2 (2.26*e*1)	3.76*e*2 (2.28*e*1)
	New method	**8.64*e*2 (2.26*e*1)**	**6.76*e*2 (2.24*e*1)**	**5.44*e*2 (2.23*e*1)**	**4.45*e*2 (2.24*e*1)**	**3.71*e*2 (2.20*e*1)**
U1000.40	Single best	7.53*e*3 (1.32*e*2)	6.44*e*3 (1.77*e*2)	5.55*e*3 (2.06*e*2)	4.79*e*3 (2.39*e*2)	4.16*e*3 (2.64*e*2)
	Simple hybrid	7.55*e*3 (1.32*e*1)	6.46*e*3 (1.77*e*2)	5.57*e*3 (2.14*e*2)	4.81*e*3 (2.48*e*2)	4.18*e*3 (2.74*e*2)
	New method	**7.48*e*3 (1.34*e*2)**	**6.40*e*3 (1.27*e*2)**	**5.50*e*3 (2.03*e*2)**	**4.76*e*3 (2.35*e*2)**	**4.14*e*3 (2.65*e*2)**

CPU: Intel Xeon E5530 2.40 GHz. Average from 1,000 runs.
Avg: average (the smaller, the better); Std: standard deviation.

Table 5: Statistical test of graph bisection.

Problem instance	Compared method	t-test	p value
G1000.2.5	Single best	+	$5.66e-10$
	Simple hybrid	+	$2.44e-03$
G1000.20	Single best	+	$2.69e-08$
	Simple hybrid	+	$5.30e-03$
U1000.05	Single best	+	$2.42e-16$
	Simple hybrid	+	$1.32e-07$
U1000.40	Single best	∼	$6.23e-02$
	Simple hybrid	+	$2.62e-04$

p value: the smaller, the more significant.
+: significantly better under level $1.00e-02$.
∼: not significantly different under level $1.00e-02$.

Table 6: Comparison of results on two test problems.

Problem instance	Our method		Galán et al. [8]				Ishibuchi and Shibata [13]			
	Avg	Std	Avg	Std	t-test	p value	Avg	Std	t-test	p value
berlin52	9.02*e*3	3.09*e*2	9.27*e*3	3.32*e*2	+	$1.6e-59$	9.24*e*3	3.38*e*2	+	$4.4e-45$
kroA100	3.51*e*4	1.48*e*3	3.81*e*4	1.60*e*3	+	$3.5e-228$	3.77*e*4	1.41*e*3	+	$9.2e-213$
bier127	1.87*e*5	6.20*e*3	1.96*e*5	6.78*e*3	+	$1.2e-158$	1.97*e*5	6.46*e*3	+	$4.4e-188$
pr152	1.76*e*5	1.00*e*4	1.96*e*5	1.12*e*4	+	$1.6e-227$	2.00*e*5	1.07*e*4	+	$3.5e-290$
G1000.2.5	2.54*e*2	8.68*e*0	3.01*e*2	3.13*e*1	+	$5.2e-248$	3.04*e*2	1.15*e*1	+	0^*
G1000.20	3.95*e*3	2.61*e*1	4.09*e*3	9.47*e*1	+	$1.0e-245$	4.09*e*3	3.21*e*1	+	0^*
U1000.05	3.71*e*2	2.20*e*1	4.96*e*2	8.40*e*1	+	$9.7e-246$	5.18*e*2	2.64*e*1	+	0^*
U1000.40	4.14*e*3	2.65*e*2	4.99*e*3	5.94*e*2	+	$5.9e-220$	5.31*e*3	2.23*e*2	+	0^*

CPU: Intel Xeon E5530 2.40 GHz. Average from 1,000 runs.
Avg: average (the smaller, the better); Std: standard deviation.
p value: the smaller, the more significant.
+: significantly better under level $1.00e-02$.
$*$: it means that this value is less than $1.0e-300$.

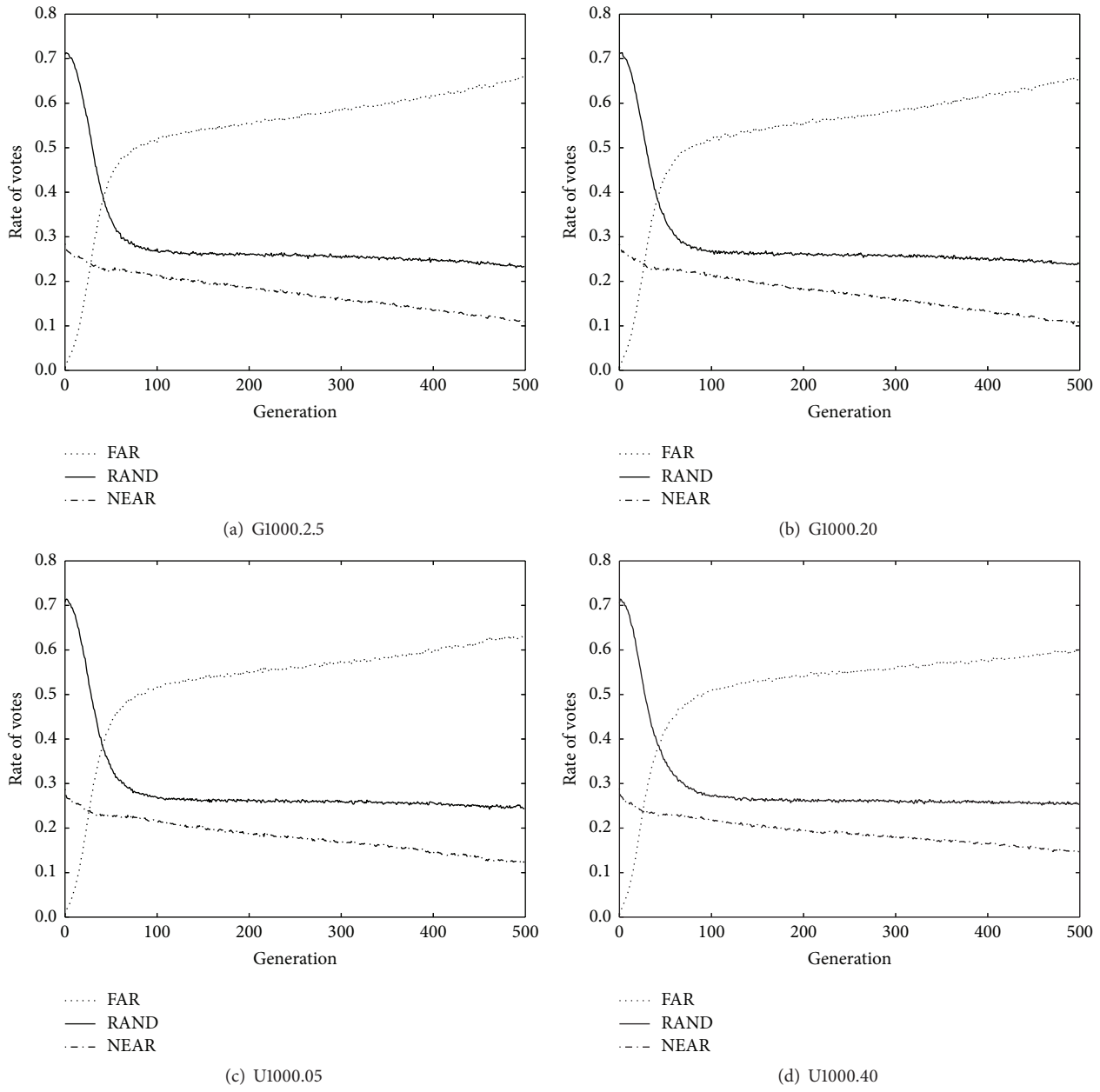


Figure 6: Voting rates of schemes in graph bisection.

with various environments such as various crossover, mutation rates, replacement, and local optimization for future work.

Disclosure

A preliminary version of this paper appeared in Proceedings of the Genetic and Evolutionary Computation Conference, 2015 (pp. 1413-1414).

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the Engineering Research Center of Excellence Program of Republic of Korea Ministry of Science, ICT\& Future Planning(MSIP)/National Research Foundation of Korea (NRF) (Grant no. NRF-0421-20120031). This research was also supported by Basic Science Research

Table 7: Results of computation time.

Problem instance	Our method			Galán et al. [8]			Ishibuchi and Shibata [13]		
	Mating	Total	Proportion of mating (%)	Mating	Total	Proportion of mating (%)	Mating	Total	Proportion of mating (%)
berlin52	7.68	81.49	9.4	0.63	70.31	0.9	68.31	140.07	48.5
kroA100	14.60	283.97	5.1	0.68	252.33	0.3	244.85	505.28	48.4
bier127	20.04	451.89	4.4	0.62	405.57	0.2	388.40	796.24	48.7
pr152	25.24	634.28	3.9	0.65	570.15	0.1	546.99	1119.84	48.8
G1000.2.5	8.26	54.81	15.0	0.60	45.94	1.3	9.06	55.74	16.2
G1000.20	7.76	61.91	12.5	0.56	52.74	1.1	9.10	63.39	14.3
U1000.05	8.98	56.45	15.9	0.57	47.02	1.2	8.86	55.22	16.0
U1000.40	8.15	66.63	12.2	0.50	57.95	1.0	9.02	68.41	13.1

Average CPU seconds from 1,000 runs on Intel Xeon E5530 2.40 GHz.

Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. 2015R1D1A1A01060105).

References

[1] G. Ochoa, C. Mädler-Kron, R. Rodriguez, and K. Jaffe, "Assortative mating in genetic algorithms for dynamic problems," in *Applications of Evolutionary Computing*, pp. 617–622, Springer, Berlin, Germany, 2005.

[2] L. B. Booker, "Improving the performance of genetic algorithms in classifier systems," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 80–92, Lawrence Erlbaum Associates, 1985.

[3] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation discussion and generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, S. Forrest, Ed., vol. 1, pp. 416–423, July 1993.

[4] F. Ramezani and S. Lotfi, "IAMGA: intimate-based assortative mating genetic algorithm," in *Proceedings of the Swarm Evolutionary and Memetic Computing Conference*, pp. 240–247, 2011.

[5] C. Fernandes, R. Tavares, C. Munteanu, and A. Rosa, "Using assortative mating in genetic algorithms for vector quantization problems," in *Proceedings of the ACM Symposium on Applied Computing (SAC '01)*, pp. 361–365, ACM, 2001.

[6] H. Ishibuchi, K. Narukawa, N. Tsukamoto, and Y. Nojima, "An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization," *European Journal of Operational Research*, vol. 188, no. 1, pp. 57–75, 2008.

[7] H. Ishibuchi and Y. Shibata, "Mating scheme for controlling the diversity-convergence balance for multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1259–1271, Seattle, Wash, USA, June 2004.

[8] S. F. Galán, O. J. Mengshoel, and R. Pinter, "A novel mating approach for genetic algorithms," *Evolutionary Computation*, vol. 21, no. 2, pp. 197–229, 2013.

[9] C. Jung, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "An adaptive method of Hungarian mating schemes in genetic algorithms," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 1413–1414, Madrid, Spain, 2015.

[10] C. Jung, Y.-H. Kim, Y. Yoon, and B.-R. Moon, "Investigation of Hungarian mating schemes for genetic algorithms," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC '14)*, pp. 140–147, ACM, Gyeongju, Republic of Korea, March 2014.

[11] D. E. Goldberg and R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, pp. 154–159, 1985.

[12] O. Martin, E. W. F. Steve, and W. Otto, "Large-step Markov chains for the traveling salesman problem," *Complex Systems*, vol. 5, no. 3, pp. 299–251, 1991.

[13] H. Ishibuchi and Y. Shibata, "A similarity-based mating scheme for evolutionary multiobjective optimization," in *Genetic and Evolutionary Computation—GECCO 2003*, E. Cantú-Paz, J. A. Foster, K. Deb et al., Eds., vol. 2723 of *Lecture Notes in Computer Science*, pp. 1065–1076, 2003.

[14] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[15] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, 1982.

[16] D. Avis, "A survey of heuristics for the weighted matching problem," *Networks*, vol. 13, no. 4, pp. 475–493, 1983.

[17] P. Hahn, T. Grant, and N. Hall, "A branch-and-bound algorithm for the quadratic assignment problem based on the Hungarian method," *European Journal of Operational Research*, vol. 108, no. 3, pp. 629–640, 1998.

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

[19] Y. Yoon, Y.-H. Kim, and B.-R. Moon, "Feasibility-preserving crossover for maximum k-coverage problem," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '08)*, pp. 593–598, Atlanta, Ga, USA, July 2008.

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

[21] K. S. Goh, A. Lim, and B. Rodrigues, "Sexual selection for genetic algorithms," *Artificial Intelligence Review*, vol. 19, no. 2, pp. 123–152, 2003.

[22] A. Moraglio and R. Poli, "Topological interpretation of crossover," in *Proceedings of the Genetic and Evolutionary*

Computation Conference (GECCO '04), vol. 3102, pp. 1377–1388, Seattle, Wash, USA, June 2004.

[23] K. A. De Jong, *An analysis of the behavior of a class of genetic adaptive systems [Ph.D. dissertation]*, University of Michigan, Ann Arbor, Mich, USA, 1975.

[24] G. Reinelt, "TSPLIB—a traveling salesman problem library," *INFORMS Journal on Computing*, vol. 3, no. 4, pp. 376–384, 1991.

[25] B. L. Welch, "The generalization of "Student's" problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1-2, pp. 28–35, 1947.

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

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2016, Article ID 4135056, 11 pages
http://dx.doi.org/10.1155/2016/4135056

Research Article

A Network Traffic Prediction Model Based on Quantum-Behaved Particle Swarm Optimization Algorithm and Fuzzy Wavelet Neural Network

Kun Zhang, Zhao Hu, Xiao-Ting Gan, and Jian-Bo Fang

School of Mathematics and Statistics, Chuxiong Normal University, Chuxiong, Yunnan 675000, China

Correspondence should be addressed to Kun Zhang; zhangkunpost@qq.com

Received 2 November 2015; Revised 14 January 2016; Accepted 3 February 2016

Academic Editor: Ahmed Kattan

Due to the fact that the fluctuation of network traffic is affected by various factors, accurate prediction of network traffic is regarded as a challenging task of the time series prediction process. For this purpose, a novel prediction method of network traffic based on QPSO algorithm and fuzzy wavelet neural network is proposed in this paper. Firstly, quantum-behaved particle swarm optimization (QPSO) was introduced. Then, the structure and operation algorithms of WFNN are presented. The parameters of fuzzy wavelet neural network were optimized by QPSO algorithm. Finally, the QPSO-FWNN could be used in prediction of network traffic simulation successfully and evaluate the performance of different prediction models such as BP neural network, RBF neural network, fuzzy neural network, and FWNN-GA neural network. Simulation results show that QPSO-FWNN has a better precision and stability in calculation. At the same time, the QPSO-FWNN also has better generalization ability, and it has a broad prospect on application.

1. Introduction

With the rapid development of computer network technology, network applications have infiltrated every corner of human society and play an important role in various industries and situations. Since the network topology structure is gradually complicated, the problem of network's emergencies and congestion are more and more serious. Through monitoring and accuracy prediction of network traffic, it can prevent network congestion and can effectively improve the utilization rate of the network [1].

In general, the network traffic data is a kind of time series data and the problem of network traffic prediction is to forecast future network traffic rate variations as precisely as possible based on the measured history. The traditional prediction model, such as Markov model [2], ARMA (Autoregressive Moving Average) model [3], ARIMA (Autoregressive Integrated Moving Average) model [4], and FARIMA (Fractional Autoregressive Integrated Moving Average) [5] model, has been proposed. As the network traffic is affected by many factors, the network traffic time series show quite obvious multiscale, long-range dependence, and nonline characteristic. The methods mentioned above have the weakness of low-level efficiency [6].

An artificial neural network (ANN) is an analysis paradigm that is roughly modeled after the massively parallel structure of the brain. Artificial neural networks can be thought of as "black box" devices that accept inputs and produce outputs and are able to give better performance in dealing with the nonlinear relationships between the output and the input theoretically [7]. Although artificial neural networks have been successfully used for modeling complex nonlinear systems and predicting signals for a wide range of engineering applications, artificial neural networks (ANNs) have limited ability to characterize local features, such as discontinuities in curvature, jumps in value or other edges [8]. These local features, which are located in time and/or frequency, typically embody important process-critical information such as aberrant process modes or faults.

The fuzzy neural networks (FNN) are the hybrid systems which combine both advantages of the fuzzy systems and artificial neural networks. The FNN possesses the merits of

the low-level learning and computational power of neural networks, and the high-level human knowledge representation and thinking of fuzzy theory [9]. A fuzzy wavelet neural network (FWNN) is a new network structure that combines wavelet theory with fuzzy logic and NNs. The synthesis of a fuzzy wavelet neural inference system includes the determination of the optimal definitions of the premise and the consequent part of fuzzy IF-THEN rules [10]. However, many fuzzy neural network models, including FWNN, have common problems derived from their fundamental algorithm [11]. For example, the design process for FNN and FWNN combined tapped delays with the backpropagation (BP) algorithm to solve the dynamic mapping problems [12]. Unfortunately, the BP training algorithm has some inherent defects [13, 14], such as low learning speed, existence of local minima, and difficulty in choosing the proper size of network to suit a given problem. Thus the systems which employ basic fuzzy inference theory make the degree of each rule extremely small and often make it underflow when the dimension of the task is large. In such a situation, the learning and inference cannot be carried out correctly.

As a variant of PSO, quantum-behaved particle swarm optimization (QPSO) is a novel optimization algorithm inspired by the fundamental theory of particle swarm and features of quantum mechanics such as the use of Schrödinger equation and potential field distribution [15]. As a global optimization algorithm, the QPSO can seek many local minima and thus increase the likelihood of finding the global minimum. This advantage of the QPSO can be applied to neural networks to optimize the topology and/or weight parameters [16].

In order to predict the network traffic more accurately, a prediction model of network traffic based on QPSO algorithm and fuzzy wavelet neural network is proposed in this paper. The network traffic data is trained by QPSO and fuzzy wavelet neural network and weights are progressively updated until the convergence criterion is satisfied. The objective function to be minimized by the QPSO algorithm is the predicted error function.

The rest of this paper is arranged as follows. Section 2 gives a brief introduction to classical PSO algorithm and quantum-behaved particle swarm optimization (QPSO) algorithm. In Section 3, the fuzzy wavelet neural network is introduced and the fuzzy wavelet neural network based on QPSO (QPSO-FWNN) algorithm is presented in detail. In Section 4, simulation results are presented. Performance metrics of the several prediction methods are analyzed and compared in Section 5. Finally, some conclusions are given in Section 6.

2. Quantum-Behaved Particle Swarm Optimization

2.1. Classical Particle Swarm Optimization. Particle swarm optimization (PSO) is an evolutionary computation technique that is proposed by Kennedy and Eberhart in 1995 [17]. Similarly to other genetic algorithms (GA), PSO is initialized with a population of random solutions. However, it is unlike GA, PSO does not have operators, such as crossover and mutation. In the PSO algorithm, each potential solution, called "particles," moves around in a multidimensional search space with a velocity constantly updated by the particle's own experience and the experience of the particle's neighbors or the experience of the whole swarm [18].

In the PSO, each particle keeps track of its coordinates in the search space which are associated with the best solution it has achieved so far and this value is called *pbest*. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population [19]. This location is called *gbest*.

The process for implementing the global version of PSO is given by the following steps.

Step 1. Initialize a population (array) of particles with random positions and velocities in the D-dimensional problem space. For a D-dimensional problem with N number of particles, the position vector X_i^t and velocity vector V_i^t are represented as

$$\begin{aligned} X_i^t &= \left[X_{i,1}^t, X_{i,2}^t, \ldots, X_{i,D}^t\right], \\ V_i^t &= \left[V_{i,1}^t, V_{i,2}^t, \ldots, V_{i,D}^t\right], \end{aligned} \tag{1}$$

where $i = 1, 2, \ldots, N$.

Step 2. For each particle, evaluate the desired optimization fitness function in D variables.

Step 3. Compare each particle's fitness evaluation with the particle's *pbest*. If the current value is better than *pbest*, then set the *pbest* value equal to the current value and the *pbest* location equal to the current location in D-dimensional space.

Step 4. Compare the fitness evaluation with the population's overall previous best. If the current value is better than *gbest*, then reset *gbest* to the current particle's array index and value.

Step 5. Update the velocity and position of the particle according to (2) and (3), respectively. One has

$$\begin{aligned} V_{i,d}^{t+1} &= V_{i,d}^t + c_1 r_1 \left(pbest_{i,d}^t - X_{i,d}^t\right) \\ &\quad + c_2 r_2 \left(gbest_d^t - X_{i,d}^t\right), \end{aligned} \tag{2}$$

$$X_{i,d}^{t+1} = X_{i,d}^t + V_{i,d}^{t+1}, \tag{3}$$

where c_1 and c_2 are two positive constants, known as the cognitive and social coefficients, which control the relative proportion of cognition and social interaction, respectively, and the values of c_1 and c_2 were decreased with each iteration [20]. r_1 and r_2 are two random values in the range $[0, 1]$. $V_{i,d}^t$, $X_{i,d}^t$, and $pbest_{i,d}^t$ are the velocity, position, and the personal best of ith particle in dth dimension for the tth iteration, respectively. The $gbest_d^t$ is the dth dimension of best particle in the swarm for the tth iteration.

Step 6. Loop to Step 2 until a stop criterion is met, usually a sufficiently good fitness or a maximum number of iteration generations.

2.2. Quantum-Behaved Particle Swarm Optimization. Motivated by concepts in quantum mechanics and particle swarm optimization, Sun et al. proposed a new version of PSO, quantum-behaved particle swarm optimization (QPSO) [21]. In the QPSO, the state of a particle is depicted by a wave function $\psi(x,t)$, instead of position and velocity. The probability density function of the particle's position is $|\psi(X,t)|^2$ in position X_i [22].

Assume that, at iteration t, particle i moves in D-dimensional space with a δ potential well centered at $pbest_{i,j}^t$ on the dth dimension. The wave function at iteration $t+1$ is given by the following equation:

$$\psi\left(X_{i,d}^{t+1}\right) = \frac{1}{\sqrt{L_{ij}^t}} \exp\left(-\frac{\left|X_{i,d}^t - pbest_{i,d}^t\right|}{L_{ij}^t}\right), \tag{4}$$

where $L_{i,d}^t$ is the standard deviation of the double exponential distribution, varying with iteration number t. Hence the probability density function Q is defined as

$$\begin{aligned} Q\left(X_{i,d}^{t+1}\right) &= \left|\psi\left(X_{i,d}^{t+1}\right)\right|^2 \\ &= \frac{1}{L_{i,d}^t} \exp\left(-\frac{2\left|X_{i,d}^{t+1} - pbest_{i,d}^t\right|}{L_{i,d}^t}\right) \end{aligned} \tag{5}$$

and the probability distribution function F is given by the following equation:

$$F\left(X_{i,d}^{t+1}\right) = 1 - \exp\left(-\frac{2\left|X_{i,d}^{t+1} - pbest_{i,d}^t\right|}{L_{i,d}^t}\right). \tag{6}$$

By using Monte-Carlo method, the jth component of position X_i at iteration $t+1$ can obtain by the following equation:

$$X_{i,d}^{t+1} = pbest_{i,d}^t \pm \frac{1}{2} L_{i,d}^t \ln\left(\frac{1}{u_{i,d}^{t+1}}\right), \tag{7}$$

where $u_{i,d}^{t+1}$ is a uniform random number in the interval $[0,1]$. The value of $L_{i,d}^{t+1}$ is calculated as

$$L_{i,d}^t = 2\alpha\left|C_d^t - X_{i,d}^t\right|, \tag{8}$$

where parameter α is known as the contraction-expansion (CE) coefficient, which can be tuned to control the convergence speed of the algorithms [23]. C^t is the mean best position (*mbest*) and is defined as

$$\begin{aligned} C^t &= \left(C_1^t, C_2^t, \ldots, C_D^t\right) \\ &= \left(\frac{1}{M}\sum_{i=1}^{M} pbest_{i,1}^t, \frac{1}{M}\sum_{i=1}^{M} pbest_{i,2}^t, \ldots, \frac{1}{M}\sum_{i=1}^{M} pbest_{i,D}^t\right), \end{aligned} \tag{9}$$

where M is the size of the population. Hence the position of the particle is updated according to the following equation:

$$X_{i,d}^{t+1} = pbest_{i,d}^t \pm \alpha\left|C_d^t - X_{i,d}^t\right| \ln\left(\frac{1}{u_{i,d}^{t+1}}\right). \tag{10}$$

From (4) and (10), the new position of the particle is calculated as

$$X_{i,d}^{t+1} = \begin{cases} pbest_{i,d}^t + \alpha\left|C_d^t - X_{i,d}^t\right| \ln\left(\dfrac{1}{u_{i,d}^{t+1}}\right) & k \in (0.5, 1], \\ pbest_{i,d}^t - \alpha\left|C_d^t - X_{i,d}^t\right| \ln\left(\dfrac{1}{u_{i,d}^{t+1}}\right) & k \in (0, 0.5], \end{cases} \tag{11}$$

where k is a random number in the range $[0,1]$. α is linearly decreasing factor from 1.0 to 0.3 with iteration as

$$\alpha^t = \alpha_{\max} - \frac{\alpha_{\max} - \alpha_{\min}}{t_{\max}} \cdot t, \tag{12}$$

where $t_{\max}$ is the maximum number iteration used in algorithm.

3. Fuzzy Wavelet Neural Network Based on QPSO

3.1. The Wavelet Base Function. In $L^2(R)$, a wavelet dictionary is constructed by dilating and translating from a wavelet base function $\psi(t)$ of zero average [24]:

$$\int_{-\infty}^{+\infty} \psi(t)\, dt = 0 \tag{13}$$

which is dilated with a scale parameter a and translated by b

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \quad a > 0. \tag{14}$$

3.2. Fuzzy Wavelet Neural Network. The basic architecture of fuzzy wavelet neural network could be described as a set of Takagi-Sugeno models. Assume that there are r rules in the rule base and the Takagi-Sugeno fuzzy if-then rules are usually in the following form:

$$\begin{aligned} &R^i\text{: IF } x_1 \text{ is } F_1^i, \ldots, x_r \text{ is } F_r^i, \\ &\text{THEN } y^i = \alpha_0^i + \alpha_1^i x_1 + \cdots + \alpha_r^i x_r, \end{aligned} \tag{15}$$

where $x_1, x_2, \ldots, x_r$ are input of T-S rule, F_j^i is the ith linguistic variable value of the jth input, which is a fuzzy set characterized by wavelet function. α_j^i is constant coefficients which are usually referred to as consequent parameters determined during the training process.

Figure 1 shows the architecture of the proposed FWNN modeling. The FWNN is a 4-layer feedforward network and detailed descriptions and equations for each layer are given here.

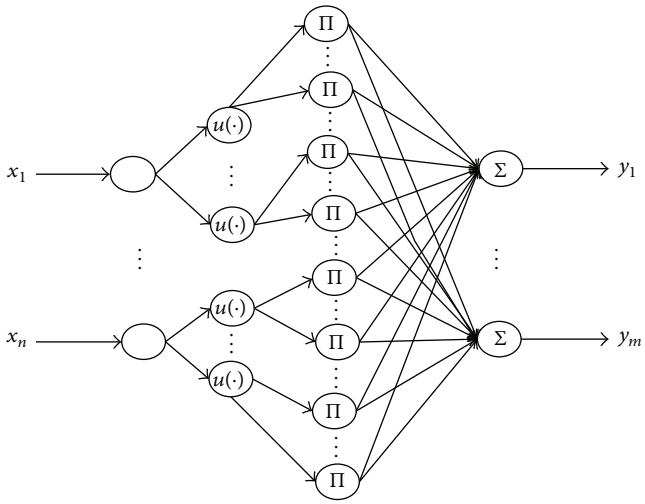


Figure 1: The architecture of FWNN.

Layer 1 (input variables layer). This layer is the input signals of the FWNN and each node of this layer, respectively, represents an input linguistic variable. The node output and the node input are related by

$$\begin{aligned} I_i^{(1)} &= x_i \\ O_i^{(1)} &= I_i^{(1)} \quad i = 1, 2, \ldots, n, \end{aligned} \tag{16}$$

where $I_i^{(1)}$ and $O_i^{(1)}$ are, respectively, the input and output of ith node in Layer 1.

Layer 2 (membership functions layer). In this layer, nodes represent fuzzy sets in the antecedents of fuzzy rules. The outputs of this layer are the values of the membership functions. The membership function is wavelet function and is often taken as

$$\begin{aligned} u_i^j &= \varphi\left(\frac{x_i - b_i^j}{a_i^j}\right) \\ &= \cos\left(\frac{x_i - b_i^j}{2a_i^j}\right) \exp\left(-\frac{1}{2}\left(\frac{x_i - b_i^j}{a_i^j}\right)^2\right), \end{aligned} \tag{17}$$

where u_i^j is the jth membership function of x_i and $j = 1, 2, \ldots, p$. a_i^j, b_i^j are the dilation and translation parameter of wavelet function.

In this layer, the relation between the output and input is represented as

$$\begin{aligned} I_{ij}^{(2)} &= O_i^{(1)}, \\ O_{ij}^{(2)} &= u_{ij}\left(I_{ij}^{(2)}\right) \\ &= \cos\left(\frac{O_i^{(1)} - b_i^j}{2a_i^j}\right) \exp\left(-\frac{1}{2}\left(\frac{O_i^{(1)} - b_i^j}{a_i^j}\right)^2\right), \end{aligned} \tag{18}$$

where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$.

Layer 3 (rule layer). In this layer, the number of rules is equal to the number of nodes. The output can be calculated as follows according to the AND (min) operation [12]:

$$O_{ij}^{(3)} = I_{ij}^{(3)} = \min\left(O_{1j}^{(2)}, O_{2j}^{(2)}, \ldots, O_{nj}^{(2)}\right). \tag{19}$$

Layer 4 (output layer). This layer consists of output nodes. The output are given by

$$I_k^{(4)} = \sum_{i=1}^{m} \sum_{j=1}^{p} O_{ij}^{(3)} \omega_{ij}^{(3)}, \tag{20}$$

$$y_k = O_k^{(4)} = \frac{I_k^{(4)}}{\sum_{i=1}^{m} \sum_{j=1}^{p} O_{ij}^{(3)}}. \tag{21}$$

To train the parameters of FWNN, backpropagation (BP) training algorithm is extensively used as a powerful training method which can be applied to the forward network architecture [25]. For this purpose, mean square error (MSE) is selected as performance index which is given by

$$J = \frac{1}{2}(D - Y)^T (D - Y), \tag{22}$$

where D and Y are current and desired output values of network, correspondingly.

The all adjustable parameters of FWNN can be calculated by the following formulas:

$$\begin{aligned} \omega_{ij}^{(3)}(t+1) &= \omega_{ij}^{(3)}(t) - \eta \cdot \frac{\partial J}{\partial \omega_{ij}^{(3)}(t)} + \alpha \cdot \Delta\omega_{ij}^{(3)}(t), \\ b_{ij}^{(2)}(t+1) &= b_{ij}^{(2)}(t) - \eta \cdot \frac{\partial J}{\partial b_{ij}^{(2)}(t)} + \alpha \cdot \Delta b_{ij}^{(2)}(t), \\ a_{ij}^{(2)}(t+1) &= a_{ij}^{(2)}(t) - \eta \cdot \frac{\partial J}{\partial a_{ij}^{(2)}(t)} + \alpha \cdot \Delta a_{ij}^{(2)}(t), \end{aligned} \tag{23}$$

where t represents the backward step number and η and α are the learning and the momentum constants, differing in the ranges 0.01 to 0.1 and 0.1 to 0.9, respectively.

3.3. Fuzzy Wavelet Neural Network Trained by QPSO Algorithm. Computational intelligence has gained popularity in training of neural networks because of their ability to find a global solution in a multidimensional search space. The QPSO algorithm is a global algorithm, which has a strong ability to find global optimistic results and QPSO algorithm has proven to have advantages than the classical PSO due its less control parameters [26]. Therefore, by combining the QPSO with the fuzzy wavelet neural network, a new algorithm referred to as QPSO-FWNN algorithm is formulated in this paper.

When QPSO algorithm is used to train the FWNN model, a decision vector represents a particular group of network

parameters including the connection weight, the dilation and translation parameter. It is further denoted as

$$X_i = \left(a_{1,1}, \ldots, a_{1,p}, \ldots, a_{n,p}, b_{1,1}, \ldots, b_{1,p}, \ldots, b_{n,p}, \omega_{1,1}, \ldots, \omega_{1,p}, \ldots, \omega_{m,p}\right), \quad (24)$$

where a_{ij} and b_{ij} are the dilation and translation parameter of wavelet function in Layer 2. $\omega_{l,k}$ are the connection weight in (20). Since a component of the position corresponds to a network parameter, FWNN is structured according the particle's position vector. Training the corresponding network by inputting the training samples, we can obtain an error value computed by (22). In a word, the mean square error is adopted as the objective function to be minimized in FWNN based on QPSO.

The specific procedure for the QPSO-FWNN algorithm can be summarized as follows.

Step 1. Define the structure of the FWNN according to the input and output sample.

Step 2. Treat the position vector of each particle as a group of network parameter by (24).

Step 3. Initialize the population by randomly generating the position vector X_i of each particle and set $pbest_i = X_i$.

Step 4. Evaluate m and α of QPSO algorithm using (9) and (12), respectively.

Step 5. Conclude the objective function of each particle by (22).

Step 6 (update *pbest*). Each particle's current fitness value is compared with previous best value *pbest*. If the current value is better than the *pbest* value, then set the *pbest* value to the current value.

Step 7 (update *gbest*). Determine the swarm best *gbest* as minimum of all the particles *pbest*.

Step 8. Judge the stopping criteria, if the maximal iterative times are met, stop the iteration, and the positions of particles are the optimal solution. Otherwise, the procedure is repeated from Step 4.

4. Simulation Results

Experimental data set consists of 10 hours observations which comes from monitoring the traffic between clients in our campus network and servers. The minimal time interval in network traffic time series is 10 seconds. Figure 2 shows the normalization of network traffic time series.

In this paper, the design of a discrete filter predictor consists in finding the relation between the future data $x(n)$ and the past observations $x(n-1), x(n-2), \ldots, x(n-M)$, where M is the number of considered input elements. The

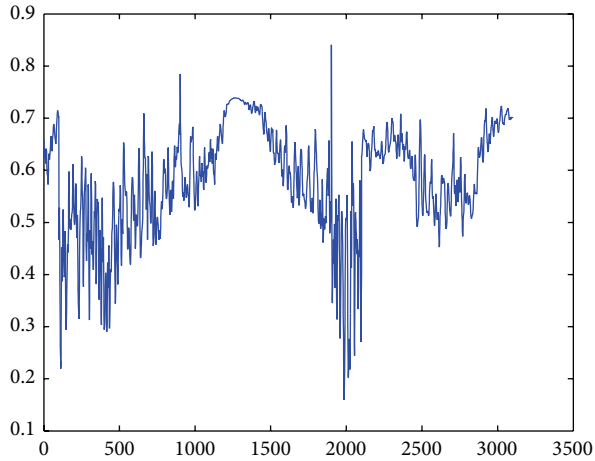


Figure 2: The normalization of network traffic time series.

predictor relationship can be described by the following convolution sum [27]:

$$\hat{x}(n) = \sum_{k=1}^{M} h_o(k)\, x(n-k), \quad (25)$$

where $h_o(k)$ $(k = 1, 2, \ldots, M)$ is the filter coefficient vector. $x(n-k)$ is an k-step backward sample. $\hat{x}(n)$ is the desired output.

In order to test the performance of the prediction model, the front 1000 is the training data, and the latter 2100 is the prediction data. The number of time series windows was set as 3, which meant that the forth of measurement data would be predicted from the past three of measurement data. In the established prediction model base on QPSO-FWNN algorithm, the number of membership functions is five, the number of input variables layer nodes is three, and the number of output layer nodes is one. The population size of QPSO algorithm is 120 particles and D-dimensional search space of particle is 45. $t_{\max}$ is 30. $\alpha_{\max}$ is 1 and $\alpha_{\min}$ is 0.5. The CE coefficient decreases linearly from 1.0 to 0.3 during the search process according to (12) and After 50 times iterations, the cost function J of QPSO-FWNN neural network was 2.1887. The connection weight between input variables layer and output layer is given by

$$a = \begin{bmatrix} 1.018 & -0.196 & -0.582 & 0.419 & 0.743 \\ -1.299 & -0.885 & 0.676 & -0.170 & 0.134 \\ 1.082 & 0.206 & 0.557 & -0.408 & 0.453 \end{bmatrix},$$

$$b = \begin{bmatrix} 0.495 & 1.510 & 1.893 & 0.869 & 2.665 \\ 1.171 & -0.627 & 2.369 & 2.304 & 0.786 \\ -0.082 & 1.297 & 2.043 & 1.300 & 1.850 \end{bmatrix},$$

$$\omega = \begin{bmatrix} -0.927 \\ 0.408 \\ 0.053 \\ 2.128 \\ 2.195 \\ 1.652 \\ 1.875 \\ 1.436 \\ -1.537 \\ 2.416 \\ 1.595 \\ 1.146 \\ 2.072 \\ 0.965 \\ 1.884 \end{bmatrix}, \tag{26}$$

where a is the dilation parameter of the wavelet function. b is the translation parameter of wavelet function. ω is the connection weight of the output layer. Figure 3 shows the membership functions of input variable $x(t)$, $x(t-1)$, and $x(t-2)$ in FWNN units. Figure 4 shows the QPSO-FWNN convergence curves. These prediction results show that the QPSO-FWNN model is an effective, high-accuracy prediction model of network traffic in Figure 5.

5. Performance Metrics

In order to evaluate the prediction model more comprehensively, the following performance metrics are used to estimate the prediction accuracy.

(1) MSE (mean square error) is a scale-dependent metric which quantifies the difference between the predicted values and the actual values of the quantity being predicted by computing the average sum of squared errors [28]:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \widehat{y}_i\right)^2, \tag{27}$$

where y_i is the actual value, $\widehat{y}_i$ is the predicted value, and N is the total number of predictions.

(2) NMSE (Normalized Mean Square Error) is defined as

$$\text{NMSE} = \frac{1}{\sigma^2}\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \widehat{y}_i\right)^2, \tag{28}$$

where σ^2 denotes the variance of the actual values during the prediction interval and is given as follows:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2, \tag{29}$$

where $\overline{y}$ is the mean value and is given by

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i. \tag{30}$$

It can be seen that, if NMSE = 0, the prediction performance is perfect, and if NMSE = 1, the prediction is a trivial predictor which statistically predicts the mean of the actual value. If NMSE > 1, it means that the prediction performance is worse than that of trivial predictor [29].

(3) MAPE (Mean Absolute Percentage Error) can calculate the prediction error as a percentage of the actual value. MAPE is defined as

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|y_i - \widehat{y}_i\right|}{y_i} \times 100\%. \tag{31}$$

(4) Coefficient correlation is the covariance of the two variables divided by the product of their individual standard deviations. It is a normalized measurement of how the two variables are linearly related. The coefficient of correlation (r) is given as follows:

$$r = \frac{\text{COV}\left(Y, \widehat{Y}\right)}{\sigma_Y \sigma_{\widehat{Y}}}, \tag{32}$$

where σ_Y and $\sigma_{\widehat{Y}}$ indicate the standard deviation of the actual and the predicted values are given by (33)

$$\sigma_Y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2}. \tag{33}$$

$\text{COV}(X, Y)$ is the covariance between X and Y. It is obtained as follows:

$$\text{COV} = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right). \tag{34}$$

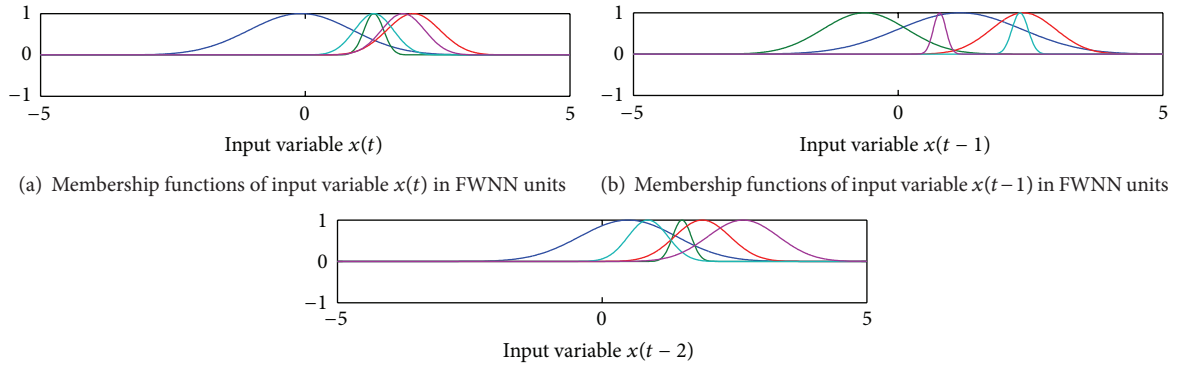
Values for the correlation coefficient range are $[-1, 1]$. If $r = 1$, there is a perfect positive correlation between the actual and the predicted values, whereas $r = -1$ indicates a perfect negative correlation. If $r = 0$, we have a complete lack of correlation among the datasets.

(5) Coefficient of efficiency (CE) is defined as

$$\text{CE} = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2}, \tag{35}$$

where the domain of the efficiency coefficient is $(-\infty, 1]$. If CE = 1, there is a perfect fit between the observed and the predicted data. When the prediction corresponds to estimating the mean of the actual values, CE = 0. If CE $\in (-\infty, 0]$, it indicates that the average of the actual values is a better predictor than the analyzed prediction method. The closer CE is to 1, the more accurate the prediction is.

In order to test the QPSO-FWNN method's validity and accuracy, we carried out the experiment which is compared with the other methods. The prediction model based on BP



(a) Membership functions of input variable $x(t)$ in FWNN units

(b) Membership functions of input variable $x(t-1)$ in FWNN units

(c) Membership functions of input variable $x(t-2)$ in FWNN units

Figure 3: The membership functions of input variable $x(t)$, $x(t-1)$, and $x(t-2)$ in FWNN units.

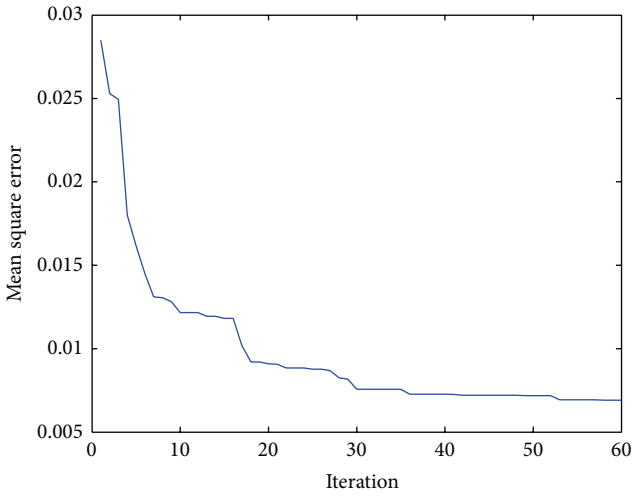


Figure 4: The QPSO-FWNN convergence curves.

Table 1: Performance comparison of the six prediction methods.

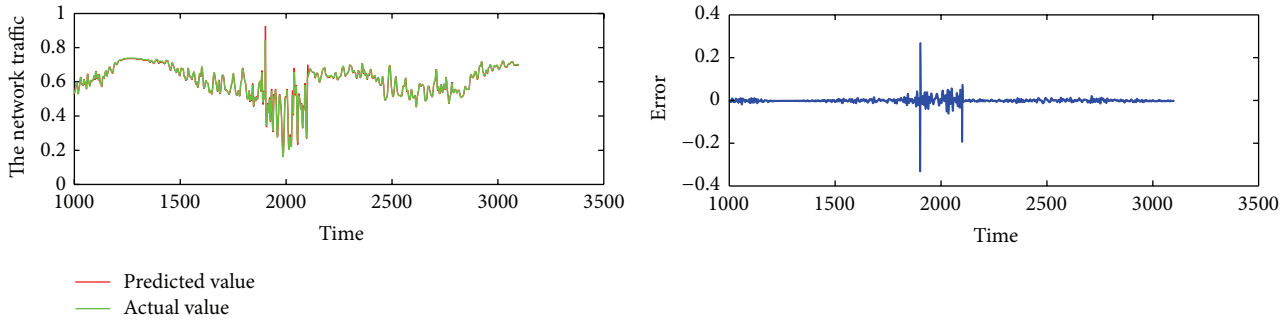
Method	MSE	NMSE	MAPE	r	CE
QPSO-FWNN	3.02×10^{-4}	0.0348	0.0181	0.9833	0.9667
BP	6.30×10^{-3}	0.3022	0.0986	0.8602	0.3091
RBF	4.20×10^{-3}	0.2182	0.0786	0.9115	0.5424
FNN	2.69×10^{-1}	1.0999	0.2041	−0.1692	−28.6128
FWNN-GA	1.01×10^{-3}	0.1475	0.0434	0.9560	0.8884
ARIMA	4.50×10^{-3}	0.5974	0.7326	0.7326	0.5011

neural network is 3 layers, in which the number of input layer nodes is 3, the number of hidden layer nodes is 7, the number of output layer nodes is 1, and the number of iterations is 1000. In the prediction model based on RBF neural network, the radial based distribute function "spread" is 1.5. The prediction model based on FNN neural network is 4 layers and the number of membership functions is five. The number of iterations is 1000. The architecture of prediction model based on fuzzy wavelet neural network and genetic algorithm (FWNN-GA) is the same with the QPSO-FWNN method. The population size of GA is 100. The crossover type is one-point crossover, and crossover rate is 0.6. Mutation rate is 0.01 and the number of iterations is 500 [30]. The prediction model based on ARIMA model is built. The estimation of the model parameters is done using Maximum Likelihood Estimation and the best model is chosen as ARIMA (5, 1, 3) [31]. Figures 6, 7, 8, 9, and 10 show that prediction results with BP, RBF, FNN, and FWNN-GA neural network and ARIMA model, respectively. The performance comparison of the four prediction methods is shown in Table 1. From Table 1, one can look further into the prediction performance among the four prediction models. By comparing the value of MSE, NMSE, MAPE, coefficient correlation, and CE, QPSO-FWNN demonstrates better prediction accuracy than the other three methods. Therefore, the experimental results in this section show that the prediction method based on QPSO-FWNN is much more effective than BP, RBF, FNN, FWNN-GA, and ARIMA. It can be seen that the prediction method based on QPSO-FWNN is a better method to predict the time series of network traffic.

In order to test the prediction stability of each mode, the five prediction methods were predicted 10 times, respectively. Figures 11, 12, and 13 show that prediction results with QPSO-FWNN models are much more stable than BP, FNN, and FWNN-GA.

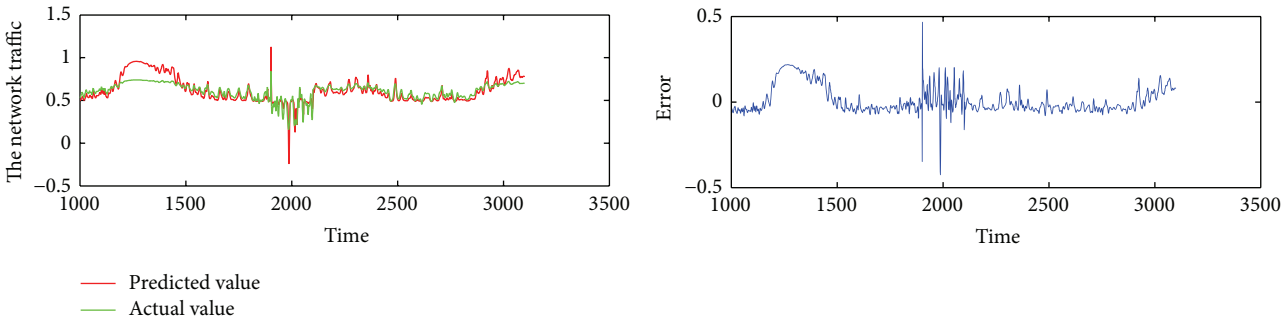
6. Conclusion

Predicting the direction of movements of network traffic is important as they enable us to detect potential network traffic jam spots. Since the network traffic is affected by many factors, the data of network traffic have the volatility and self-similarity features and the network traffic prediction becomes a challenge task. In this paper, the QPSO-FWNN method has been presented to predict the network traffic. The QPSO-FWNN combines the QPSO, which has the merit of powerful global exploration capability, with FWNN which can extract the mapping relations between the input and output data. The parameters of FWNN neural network are obtained by quantum-behaved particle swarm optimization (QPSO) and the time series of network traffic data was modeled by QPSO-FWNN. Finally, experiments showed that QPSO-FWNN model has faster and better performance in



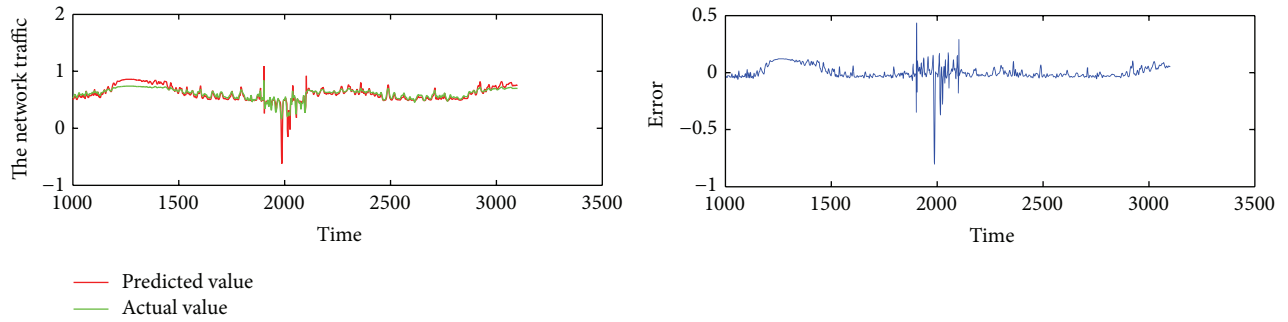
(a) The predicted results with QPSO-FWNN method (b) The error between the actual value and its predicted value

Figure 5: Prediction with QPSO-FWNN model.



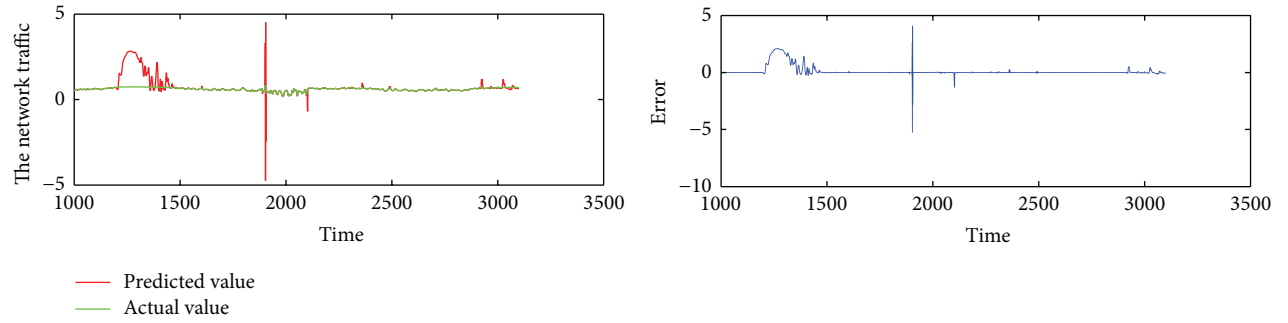
(a) The predicted results with BP neural network (b) The error between the actual value and its predicted value

Figure 6: Prediction results with BP neural network.



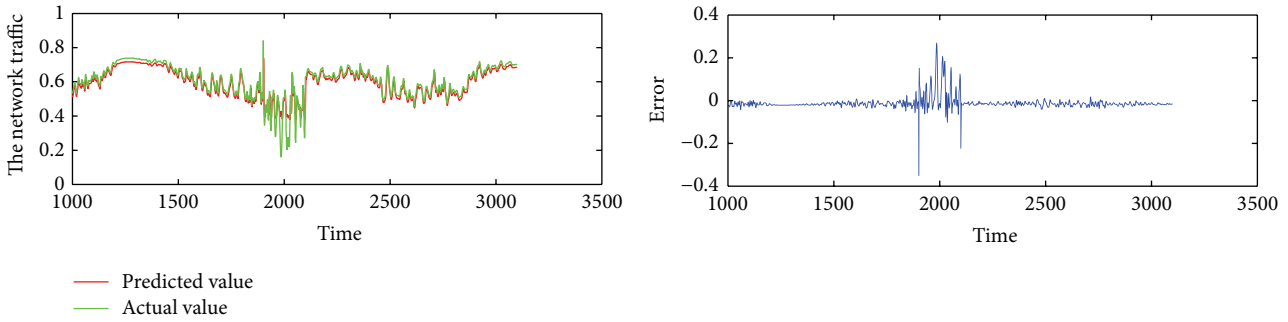
(a) The predicted results with RBF method (b) The error between the actual value and its predicted value

Figure 7: Prediction results with RBF neural network.



(a) The predicted results with FNN method (b) The error between the actual value and its predicted value

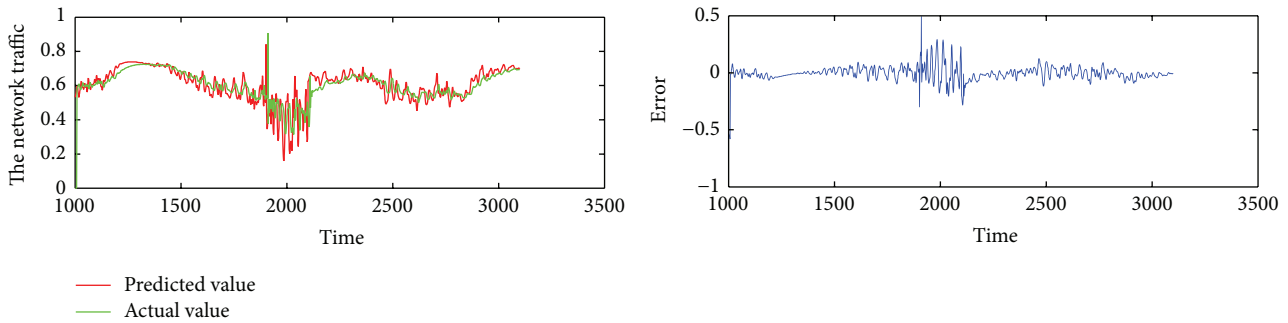
Figure 8: Prediction results with FNN neural network.



(a) The predicted results with FWNN-GA method

(b) The error between the actual value and its predicted value

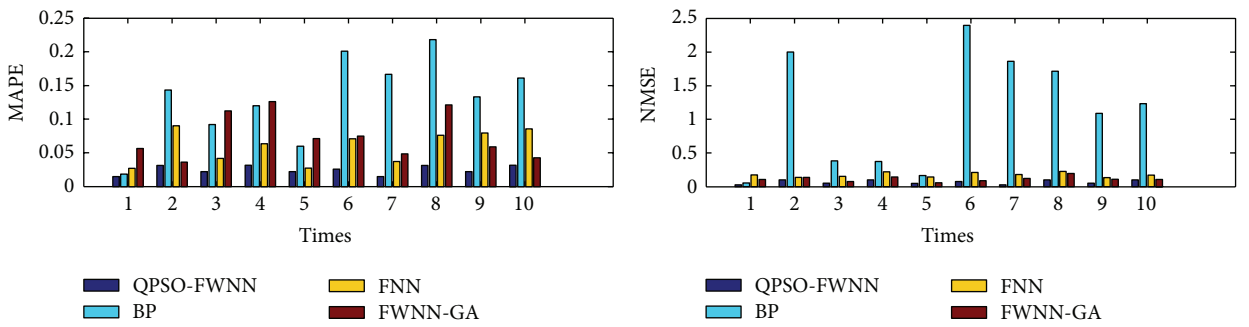
Figure 9: Prediction results with FWNN-GA method.



(a) The predicted results with ARIMA (5, 1, 3) model

(b) The error between the actual value and its predicted value

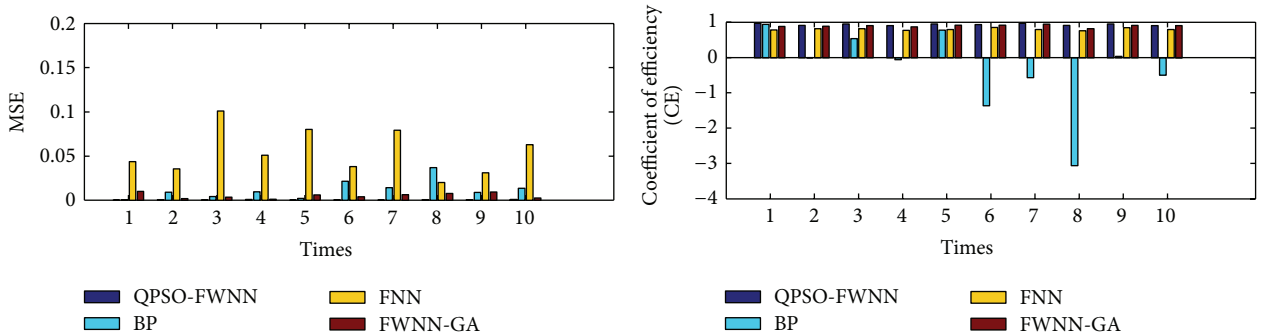
Figure 10: Prediction results with ARIMA method.



(a) MAPE histogram of four prediction models

(b) NMSE histogram of four prediction models

Figure 11: MAPE and NMSE histogram of four prediction models.



(a) MSE histogram of five prediction models

(b) CE histogram of five prediction models

Figure 12: MSE and CE histogram of four prediction models.

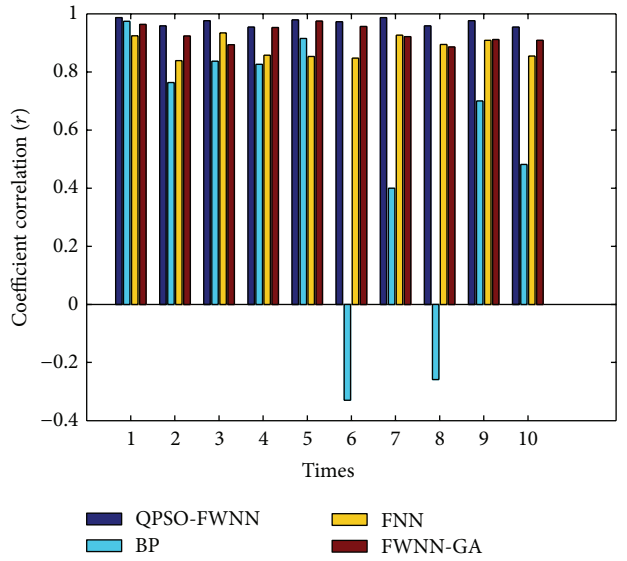


Figure 13: Coefficient correlation histogram of four prediction models.

prediction of nonlinear and nonstationary time series than many pure neural networks.

Conflict of Interests

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

Acknowledgment

This research was partially supported by the National Natural Science Foundation of China (no. 11261001).

References

[1] H. Mahmassani, "Dynamic network traffic assignment and simulation methodology for advanced system management applications," *Networks and Spatial Economics*, vol. 1, no. 3, pp. 267–292, 2001.

[2] A. Nogueira, P. Salvador, R. Valadas et al., "Markovian modelling of internet traffic," in *Network Performance Engineering*, vol. 5233 of *Lecture Notes in Computer Science*, pp. 98–124, Springer, Berlin, Germany, 2011.

[3] S. Y. Chang and H.-C. Wu, "Novel fast computation algorithm of the second-order statistics for autoregressive moving-average processes," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 526–535, 2009.

[4] M.-C. Tan, S. C. Wong, J.-M. Xu, Z.-R. Guan, and P. Zhang, "An aggregation approach to short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 60–69, 2009.

[5] Y. Shu, Z. Jin, L. Zhang, L. Wang, and O. W. W. Yang, "Traffic prediction using FARIMA models," in *Proceedings of the IEEE International Conference on Communications (ICC '99)*, vol. 2, pp. 891–895, IEEE, Vancouver, Canada, June 1999.

[6] R. Li, J.-Y. Chen, Y.-J. Liu, and Z.-K. Wang, "WPANFIS: combine fuzzy neural network with multiresolution for network traffic prediction," *Journal of China Universities of Posts and Telecommunications*, vol. 17, no. 4, pp. 88–93, 2010.

[7] K. Zhao, "Global robust exponential synchronization of BAM recurrent FNNs with infinite distributed delays and diffusion terms on time scales," *Advances in Difference Equations*, vol. 2014, article 317, 2014.

[8] E. A. Rying, G. L. Bilbro, and J.-C. Lu, "Focused local learning with wavelet neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 304–319, 2002.

[9] D. Lin, X. Wang, F. Nian, and Y. Zhang, "Dynamic fuzzy neural networks modeling and adaptive backstepping tracking control of uncertain chaotic systems," *Neurocomputing*, vol. 73, no. 16–18, pp. 2873–2881, 2010.

[10] R. H. Abiyev and O. Kaynak, "Fuzzy wavelet neural networks for identification and control of dynamic plants—a novel structure and a comparative study," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 8, pp. 3133–3140, 2008.

[11] H. Iyatomi and M. Hagiwara, "Adaptive fuzzy inference neural network," *Pattern Recognition*, vol. 37, no. 10, pp. 2049–2057, 2004.

[12] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2000.

[13] W. Li and Y. Hori, "An algorithm for extracting fuzzy rules based on RBF neural network," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1269–1276, 2006.

[14] G. K. Venayagamoorthy, "Online design of an echo state network based wide area monitor for a multimachine power system," *Neural Networks*, vol. 20, no. 3, pp. 404–413, 2007.

[15] W. Fang, J. Sun, and W. Xu, "A new mutated quantum-behaved particle swarm optimizer for digital IIR filter design," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, Article ID 367465, 7 pages, 2009.

[16] L. dos Santos Coelho and V. C. Mariani, "Particle swarm approach based on quantum mechanics and harmonic oscillator potential well for economic load dispatch with valve-point effects," *Energy Conversion and Management*, vol. 49, no. 11, pp. 3080–3085, 2008.

[17] L. D. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1676–1683, 2010.

[18] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," *Journal of Artificial Evolution and Applications*, vol. 2008, Article ID 685175, 10 pages, 2008.

[19] R. C. Eberhart and Y. Shi, *Computational Intelligence: Concepts to Implementations*, Elsevier, Philadelphia, Pa, USA, 2009.

[20] S. L. Sabat, L. dos Santos Coelho, and A. Abraham, "MESFET DC model parameter extraction using Quantum Particle Swarm Optimization," *Microelectronics Reliability*, vol. 49, no. 6, pp. 660–666, 2009.

[21] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '04)*, pp. 325–331, Portland, Ore, USA, June 2004.

[22] 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.

[23] J. Sun, W. Fang, V. Palade, X. Wu, and W. Xu, "Quantum-behaved particle swarm optimization with Gaussian distributed

local attractor point," *Applied Mathematics and Computation*, vol. 218, no. 7, pp. 3763–3775, 2011.

[24] S. Mallate, *A Wavelet Tour of Signal Processing: The Sparse Way*, Elsevier, Philadelphia, Pa, USA, 2009.

[25] F.-J. Lin, C.-H. Lin, and P.-H. Shen, "Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 5, pp. 751–759, 2001.

[26] S. M. Mikki and A. A. Kishk, "Quantum particle swarm optimization for electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 10, pp. 2764–2775, 2006.

[27] F. H. T. Vieira, G. R. Bianchi, and L. L. Lee, "A network traffic prediction approach based on multifractal modeling," *Journal of High Speed Networks*, vol. 17, no. 2, pp. 83–96, 2010.

[28] M. Barabas, G. Boanea, A. B. Rus, V. Dobrota, and J. Domingo-Pascual, "Evaluation of network traffic prediction based on neural networks with multi-task learning and multiresolution decomposition," in *Proceedings of the IEEE 7th International Conference on Intelligent Computer Communication and Processing (ICCP '11)*, pp. 95–102, Cluj-Napoca, Romania, August 2011.

[29] Y. Liang, "Real-time VBR video traffic prediction for dynamic bandwidth allocation," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 34, no. 1, pp. 32–47, 2004.

[30] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, "Hybrid taguchi-genetic algorithm for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 365–377, 2004.

[31] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1110–1124, 2005.