Research Article

A Variable Neighborhood MOEA/D for Multiobjective Test Task Scheduling Problem

Hui Lu, Zheng Zhu, Xiaoteng Wang, and Lijuan Yin

School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

Correspondence should be addressed to Hui Lu; mluhui@buaa.edu.cn

Received 26 October 2013; Revised 2 January 2014; Accepted 19 February 2014; Published 1 April 2014

Academic Editor: Kui Fu Chen

Copyright © 2014 Hui Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Test task scheduling problem (TTSP) is a typical combinatorial optimization scheduling problem. This paper proposes a variable neighborhood MOEA/D (VNM) to solve the multiobjective TTSP. Two minimization objectives, the maximal completion time (makespan) and the mean workload, are considered together. In order to make solutions obtained more close to the real Pareto Front, variable neighborhood strategy is adopted. Variable neighborhood approach is proposed to render the crossover span reasonable. Additionally, because the search space of the TTSP is so large that many duplicate solutions and local optima will exist, the Starting Mutation is applied to prevent solutions from becoming trapped in local optima. It is proved that the solutions got by VNM can converge to the global optimum by using Markov Chain and Transition Matrix, respectively. The experiments of comparisons of VNM, MOEA/D, and CNSGA (chaotic nondominated sorting genetic algorithm) indicate that VNM performs better than the MOEA/D and the CNSGA in solving the TTSP. The results demonstrate that proposed algorithm VNM is an efficient approach to solve the multiobjective TTSP.

1. Introduction

During recent decades, the manufacturing of electronic devices has become highly integrated and increasingly complex. As a result, the resource and time consumption expended on the test of electronic devices became a crucial problem in engineering application. Therefore, the research for improving the test efficiency is a topic that has attracted extensive attention. To address this situation, the objective of this research is to solve the test task scheduling problem (TTSP) more efficiently.

The goal of the TTSP is to arrange the execution of n tasks on m instruments. It is a difficult nondeterministic polynomial (NP) problem [1] for optimization. TTSP has some similarities with flexible job shop scheduling problem (FJSP) [2, 3], but the resource configuration of the TTSP is more flexible. For example, in the TTSP, one task can be performed on more than one instrument at a time. The precedence relationships in the TTSP resemble a network. One task can have one or more former or latter tasks in the TTSP. Generally speaking, feasible solutions are more difficult to be obtained in the TTSP than that in the FJSP.

TTSP, FJSP, and most scheduling problems belong to combinatorial optimization problems. For combinatorial optimization problems, the search space is too large that the best solution cannot be obtained by adopting the method of enumeration for even small-scale problem. Therefore, the intelligent algorithms based on integer programming model are devoted to solving these kinds of problems. Take FJSP as the example; genetic algorithm (GA) [4–7], simulated annealing (SA) [8–10], and the tabu search (TS) [11] have been successfully applied in solving scheduling optimization problem. FJSP receives extensive attention and researches, and many hybrid intelligent algorithms are invented for improving the performance of the solutions. For example, a combination of shuffled frog leaping and fuzzy logic is proposed to solve FJSP [12]. A particle swarm optimization (PSO) algorithm and TS algorithm are combined to solve the multiobjective FJSP [13]. A biogeography-based optimization (BBO) algorithm [14] was proposed for FJSP for finding optimum or near-optimum solution. Hybrid discrete particle swarm optimization for multiobjective flexible job-shop scheduling problem was proposed in article [15] especially for large-scale problems. The objective functions are different in

each literature, but the makespan, the total tardiness, the critical machine workload, and the total workload of machines are frequently considered factors in those researches.

Different from the research of FJSP, the research of TTSP is relatively few because of the development of automatic test system. However, there are still some achievements in TTSP. Xia et al. [16] proposed a method that combined GA and simulated SA to optimize the parallel efficiency and speed up ratio of the multi-UUT parallel test. Genetic Algorithm-Ant Colony Algorithm (GA-ACA) [17], hybrid particle swarm and tabu search [18] and Ant Colony Algorithm [19] are used to solve parallel test tasks scheduling to obtain the minimum makespan. A chaotic nondominated sorting genetic algorithm is proposed to solve multiobjective TTSP [20]. The chaotic operations are combined with NSGA-II [21] in this approach. Those algorithms have shown excellent property in decreasing costs and improving efficiencies in automatic test system.

There are also some intelligent algorithms used to solve power dispatch problems and other scheduling problems. For example, opposition-based learning is employed in opposition-based gravitational search algorithm (OGSA) to solve optimal reactive power dispatch [22]. A fuzzified multiobjective PSO (FMOPSO) algorithm is proposed and implemented to dispatch the electric power [23]. An interactive artificial bee colony algorithm was proposed for the multiobjective environmental/economic dispatch problem [24].

In summary, most of the researches of scheduling problems focus on the single-objective problem or adopt weighted sum approach to convert the multiobjective problem into a single-objective problem. However, the weighting coefficients are difficult to choose, and human factors will greatly impact the performance of the algorithms. In fact, there are another two kinds of methods for solving the multiobjective problem. One method is the non-Pareto approach utilizing operators for processing the different objectives in a separated way. Another is the Pareto approaches which are directly based on the Pareto optimality concept. They aim at satisfying two goals: converging towards the Pareto front and also obtaining diversified solutions scattered all over the Pareto front. Those two kinds of methods mainly rely on the performance and strategies of the algorithms used in the multiobjective problems.

In this paper, the method based on Tchebycheff decomposition for multiobjective functions was adopted and the algorithm named MOEA/D is used to solve TTSP. MOEA/D is a typical evolutionary algorithm based on decomposition proposed by Zhang and Li [25]. This method decomposes a multiobjective optimization problem into a number of scalar optimization subproblems and optimizes them simultaneously. The results show that MOEA/D has a good performance for the ZDT and DTLZ test problems. MOEA/D is very efficient in solving multiobjective problems. Research on MOEA/D has also been performed in recent years. For example, Tan et al. [26] proposed a new version of MOEA/D with a uniform design to deal with the multiobjective problem in higher-dimensional objective spaces. This

method can render the distribution of the weighting vectors more uniform, especially for problems with high dimension. Chen et al. [27] introduced a guided mutation operator and priority update to enhance the ability of MOEA/D. Stochastic ranking and constraint-domination principle are adopted in MOEA/D to improve the ability of the algorithm to deal with constrained multiobjective optimization problems [28]. Although these studies have improved the ability of MOEA/D for solving multi-objective problems, MOEA/D is mainly used to solve standard test cases like ZDT, DTLZ, and FI. However, MOEA/D is rarely used to solve combinational optimization problems such as FJSP, TTSP. Peng et al. applied MOEA/D to solve Travelling Salesman Problem (TSP) [29]. However, there is no special improvement for MOEA/D according to the feature of MOEA/D and the property of TSP.

The scheduling problems, such as TTSP, FJSP, and TSP, and power dispatching problem are a branch of combinational optimization problems. Because of the properties of the combinational optimization problems, the final best solutions only account for a rather small subset of the search space. How to avoid the solutions obtained being trapped in local optima is the key to improve the ability of algorithms to deal with combinational optimization problems. Considering the fact that the size of the neighborhood is important in MOEA/D [25], too large size will lead to degradation and too small size will weaken the effect of evolutionary process. Moreover, there will be many duplicate solutions due to the influence of neighborhood updating of MOEA/D [25]. The population diversity will decrease obviously. Based on the analyses above, variable neighborhood based on a quadratic curve is adopted to ensure that the crossover span is more reasonable, and Gauss mutation is adopted at the beginning of iteration to maintain the diversity of the population. These two improvements can efficiently enhance the ability of MOEA/D for avoiding the solutions obtained from being trapped in local optima. The proposed approach cannot only solve TTSP but also deal with other scheduling problems, because the feasible solutions of TTSP are more difficult to obtain than most scheduling problems such as FJSP, TSP.

The organization of this paper is as follows. A brief introduction of TTSP is introduced in Section 2. The new method for TTSP, variable neighborhood MOEA/D (VNM), is proposed in Section 3. The convergence analysis of VNM is also presented in Section 4. A large number of experimental results and discussions are covered in Section 5. Conclusions are given in Section 6.

2. The Formulation of TTSP

2.1. The Mathematical Model for TTSP. The goal of the TTSP is to arrange the execution of n tasks on m instruments. There are three main mathematical models for TTSP. One model is based on Petri net. The second is based on Graph theory. And the third model is based on integer programming. Our work in this paper is mainly based on the integer programming proposed by us in paper [20].

FIGURE 1: The Petri net model for one task TTSP.

2.1.1. The Petri Net Model for TTSP. Petri net [30, 31] was proposed in 1962. Petri net focuses on the changes of the system, the conditions for changes, the influence of changes, and the relationships between changes. We assume that there is one test task t_1 in TTSP. The instruments occupied for t_1 are r_1 , r_2 , and r_3 . The Petri net model for this TTSP can be shown as Figure 1. In this model, there are four places (p_1 , p_2 , p_3 , and p_4), one transition (t_1), three tokens (r_1 , r_2 , and r_3), three variables (v_1 , v_2 , and v_3), four arc expressions (v_1 , v_2 , v_3 , and (v_1, v_2, v_3)), and a guard ($[r_1, r_2, r_3]$), where v_1 , v_2 , and v_3 are bound to r_1 , r_2 , and r_3 .

In Figure 1, at the beginning, test resources r_1 , r_2 , and r_3 are vacant. The corresponding tokens for three places p_1 , p_2 , and p_3 are r_1 , r_2 , and r_3 , respectively. Therefore, r_1 , r_2 , and r_3 can be allocated to t_1 . When the t_1 is finished, the tokens in p_1 , p_2 , and p_3 will be transferred to place p_4 . The tokens in p_4 are v_1 , v_2 , and v_3 . This means that resources r_1 , r_2 , and r_3 are released. The Petri net can describe the relationships between tasks by the places and transitions, but the complex models are needed to be established. The process will increase the development cost and extend the development cycle.

2.1.2. The Graph Theory Model for TTSP. Graph theory [32] is an important branch of mathematics. By adopting the Graph theory, the complex project planning and processing can be described using “graphs.” In TTSP, the vertexes of the graph represent the test tasks, and the lines between vertexes mean that some test instruments are common for these two tasks. For example, there are four test tasks (t_1 , t_2 , t_3 , and t_4) and four test instruments (r_1 , r_2 , r_3 , and r_4). The instruments set needed by t_1 , t_2 , t_3 , t_4 are $\{r_1, r_2\}$, $\{r_2, r_4\}$, $\{r_3, r_4\}$, and $\{r_1, r_3\}$, respectively. The graph for this TTSP example is shown in Figure 2.

Graph theory model can only be adopted by typical optimization methods. With the increment of the scale of TTSP, the computation expense will greatly increase, but

FIGURE 2: The Graph model for TTSP.

typical optimization methods are not suitable for large-scale TTSP problem. Therefore, Graph theory model cannot solve large-scale TTSP also.

2.1.3. The Integer Programming Model for TTSP. TTSP is a typical integer programming problem. For the integer programming model for TTSP, the TTSP can be described as follows [20]: assume that n tasks and m instrument are included in TTSP. There is a task set $T = \{t_1, t_2, \dots, t_j\}$ ($1 \leq j \leq n$) and an instrument set $R = \{r_1, r_2, \dots, r_i\}$ ($1 \leq i \leq m$). S_j^i , C_j^i , and P_j^i represent the test start time, test finish time, and test consumed time of task t_j tested on instrument r_i , respectively. In the TTSP, one task can be tested on more than one instrument. A judgment matrix is used to express whether instrument r_i is needed for t_j . The judgment matrix is defined as the following:

$$O_j^i = \begin{cases} 1 & \text{if } t_j \text{ occupies } r_i \\ 0 & \text{others.} \end{cases} \quad (1)$$

In general, task t_j may have several possible test schemes. The set of test schemes for t_j is defined as $W_j = \{w_j^1, w_j^2, \dots, w_j^{k_j}\}$ (k_j is the number of test schemes for t_j). The notation $P_j^k = \max_{r_i \in w_j^k} P_j^i$ is used to express the test time of t_j for w_j^k .

The following describes the restriction of resources:

$$X_{jj}^{kk*} = \begin{cases} 1 & \text{if } w_j^k \cap w_{j*}^{k*} \neq \emptyset \\ 0 & \text{others.} \end{cases} \quad (2)$$

Basic hypothesis includes three factors. At a given time, an instrument can only execute one task; each task must be completed without interruption once it starts. Assume $P_j^i = P_j^k$, $C_j^i = S_j^i + P_j^i$ to simplify the problem.

2.2. The Objective Functions for TTSP. The objective functions are very important in the study of multiobjective optimization problem. The makespan is very important in scheduling problems such as TTSP and FJSP, because the completion time is an essential factor for scheduling problem in product process. In additional, for TTSP, the test instruments have high integration, and the test instruments have become increasingly expensive. Therefore, the demand

for reducing the workload of the instruments and increasing the service life of the test instruments has great significance in TTSP. Therefore, our work focuses on two main objectives. One is to minimize the maximal test completion time, and the other is to minimize the mean workload of the instruments. These objectives are represented by $f_1(x)$ and $f_2(x)$.

(1) *The Maximal Test Completion Time $f_1(x)$.* The notation $C_j^k = \max_{r_i \in w_j^k} C_j^i$ is the test completion time of t_j for w_j^k . Thus, the maximal test completion time of all tasks can be defined as follows:

$$f_1(x) = \max_{1 \leq k \leq k_j, 1 \leq j \leq n} C_j^k. \quad (3)$$

(2) *The Mean Workload of the Instruments $f_2(x)$.* First, a new notation Q is introduced to describe the parallel steps. The initial value of Q is 1. Assign the instruments for all of the tasks, if $X_{jj}^{kk*} = 1$, $Q = Q + 1$. Therefore, the mean workload of the instruments can be defined as follows:

$$f_2(x) = \frac{1}{Q} \sum_{j=1}^n \sum_{i=1}^m P_j^i O_j^i. \quad (4)$$

3. The Variable Neighborhood MOEA/D Algorithm

In this section, we proposed a variable neighborhood MOEA/D algorithm (VNM). To obtain solutions close to the real Pareto Front (PF) of the TTSP, two strategies are adopted. The variable neighborhood strategy helps to make the crossover span more reasonable. Moreover, Gauss mutation is adopted at the beginning of the iteration to maintain the diversity of the population.

3.1. The Main Strategy of the VNM. The VNM is an evolutionary algorithm based on decomposition. The main strategy of the VNM is to decompose a multiobjective optimization problem into a number of scalar optimization subproblems and optimize these subproblems simultaneously. The decomposition method used is the Tchebycheff approach [33]. Each subproblem is bound with a weight vector, and then each subproblem is updated by obtaining information from its neighborhood [25]. The neighborhood of each subproblem is determined by its weighting vector.

Let $\{\lambda^1, \lambda^2, \dots, \lambda^N\}$ be a set of weight vectors, and $z^* = (z_1^*, z_2^*, \dots, z_m^*)^T$ is defined as the reference point. The problem of the Pareto Front approximation can be decomposed into N scalar optimization subproblems using the Tchebycheff approach, and the objective function of the j th subproblem is defined as

$$g^{te}(x | \lambda^j, z^*) = \max_{1 \leq i \leq m} \{\lambda_i^j |f_i(x) - z_i^*|\}, \quad x \in \Omega, \quad (5)$$

FIGURE 3: The main procedure of the VNM.

where Ω is the decision space and $\lambda^j = (\lambda_1^j, \lambda_2^j, \dots, \lambda_m^j)^T$, $z_i^* = \min\{f_i(x) | x \in \Omega\}$ for each $i = 1, 2, \dots, m$. It is clear that the VNM is able to minimize all N objective functions simultaneously in a single run.

The main procedure of the VNM can be described as shown in Figure 3.

In the part of parameter setting, the iteration number M , the subproblem number N , the size of neighborhood T (which ranges from beginning size B to stopping size S), and the population for saving the optimal solutions EP are set.

The crossover operation in VNM is as follows.

For each individual x_i^t in generation t , the child x_i^{t+1} can be obtained by the following equation:

$$x_i^{t+1} = \begin{cases} x_i^t + F_1 \times (x_i^t - x_{i1}^t) + F_2 \times (x_i^t - x_{i2}^t) & \text{rand}(1) < CR \\ x_i^t & \text{rand}(1) \geq CR \end{cases} \quad (6)$$

CR , F_1 , and F_2 are the three control variables for the crossover; x_{i1}^t and x_{i2}^t are two individuals chosen in the neighborhood of x_i^t . This crossover method can make full use of the information from the neighborhood and render the information exchange more sufficient.

The main idea of VNM is given above. Two improvements are involved in the VNM algorithm. Variable neighborhood strategy is adopted to make the crossover span more reasonable. Moreover, Starting Mutation is used to enhance the diversity of the population.

FIGURE 4: Three controlling curves for the neighborhood size.

3.2. Variable Neighborhood. In the VNM, the size of the neighborhood T has a high impact on the performance of the algorithm. If T is too large, the two solutions chosen (x^l and x^k) for the genetic operation may be unsuitable for the subproblem, and degradation may occur during the progress of the evolution. In contrast, if T is too small, the subproblems are all similar. The child individual will be so similar to its parents that the crossover operation will have a weak effect.

T is the neighborhood size which determines the crossover and neighborhood updating span. Too large and too small T will both have a negative influence on VNM. Therefore, T should be large enough at the beginning of the evolution period to ensure sufficient information exchange of the solutions, and T should be sufficiently small in the latter portion of the evolution period such that degradation can be avoided. Motivated by this ideology, we designed and tested three curves to find the best T controlling curve.

The three curves are shown in Figure 4. In this figure, the abscissa is the number of iterations, and the ordinate is the size of the neighborhood. $M1$, $M2$, and $M3$ represent the straight line, the monotonic parabolic and the nonmonotonic parabolic curves, respectively. It is worth noting that in curve $M2$, the curvature will be 0 at the end of the evolution period. This means that the rate of change of curvature for $M2$ is the fastest of all of the concave monotone parabolas during the period of evolution. Because the curvature goes to 0 in the end, curve $M2$ is determined. Assume that if the number of iterations is 125, the neighborhood of curves $M1$, $M2$, and $M3$ are y_1 , y_2 , and y_3 , respectively, in accordance with the equation: $y_1 - y_2 = y_2 - y_3$. Thus, curve $M3$ can be also determined. Curve $M3$ is a nonmonotonic parabolic curve. A series of experiments should be performed to compare the influence of the three curves on the algorithm to identify the best controlling curve.

3.3. Starting Mutation. The TTSP represents a typical combinatorial optimization problem. The final best solutions may be limited to only several points in the solution space. Because of the neighborhood updating effect of the VNM, there will

be many duplicate solutions so that the crossover operation will have little effect. Therefore, how to maintain the diversity of the population is the key question for enhancing the algorithm effect.

Motivated by the ideology above, a starting Gauss mutation is adopted at the beginning of the iteration. For a solution $x^i = (x_1^i, x_2^i, \dots, x_M^i)$ (M is the number of variables), Gauss mutation is described as the following:

$$\text{for } j = 1, 2, \dots, M, x_j^{i*} = \begin{cases} \text{normal}(x_j^i, \sigma) & \text{rand}(1) < p \\ x_j^i & \text{rand}(1) \geq p. \end{cases} \quad (7)$$

$x^{i*} = (x_1^{i*}, x_2^{i*}, \dots, x_M^{i*})$ represents the individual after mutation, p is the mutation probability, $\text{normal}(x_j^i, \sigma)$ is a number that obeys the normal distribution, x_j^i is the mean value, and σ is the variance. With Starting Mutation, the problem with the initially invalid crossover operation can be resolved. Therefore, we can avoid the solutions from becoming trapped in local optima, and thus solutions with higher quality are obtained.

4. The Convergence Analysis of VNM

The convergence analysis of VNM in this section provides the theory ground for its application. The convergence behavior of VNM is analyzed according to the Markov Chain and the transfer matrix, respectively.

4.1. Strong and Weak Convergence. This section proposes the basic theories of convergence and proves the strong and weak convergence of VNM from the perspective of Markov Chain.

There is a global optimal solution set M for MOPs (multiobjective problem). M is defined as $M = \{X; \forall Y \in S, f(X) \geq f(Y)\}$. It is assumed that $\vec{X}(n)$ is the population in evolutionary algorithms.

A detailed demonstration for the convergence of MOEA has been proposed in paper [34]. Based on it, the definitions are described as follows.

Theorem 1. α_n , β_n , and r_n are defined as

$$\begin{aligned} \alpha_n &= P \{ \vec{X}(n+1) \cap M = \emptyset / \vec{X}(n) \cap M \neq \emptyset \}, \\ \beta_n &= P \{ \vec{X}(n+1) \cap M = \emptyset / \vec{X}(n) \cap M = \emptyset \}, \\ r_n &= \beta_1 \beta_2 \sim \beta_n. \end{aligned} \quad (8)$$

If $\lim_{n \rightarrow \infty} r_n = 0$, $\vec{X}(n)$ converges to global optimal solution weakly. It is defined as $\vec{X}(n) \rightarrow M(P.W.)$.

Theorem 2. $\bar{\alpha}_n$, $\bar{\beta}_n$, and $\bar{r}_n$ are defined as

$$\begin{aligned} \bar{\alpha}_n &= P \{ \vec{X}(n+1) \cap M^c \neq \emptyset / \vec{X}(n) \cap M^c = \emptyset \}, \\ \bar{\beta}_n &= P \{ \vec{X}(n+1) \cap M^c \neq \emptyset / \vec{X}(n) \cap M^c \neq \emptyset \}, \\ \bar{r}_n &= \bar{\beta}_1 \bar{\beta}_2 \sim \bar{\beta}_n. \end{aligned} \quad (9)$$

If $\lim_{n \rightarrow \infty} \bar{r}_n = 0$, $\vec{X}(n)$ converges to global optimal solution strongly. It is defined as $\vec{X}(n) \rightarrow M(P.S.)$.

Based on Theorems 1 and 2 above, the demonstration for the convergence of VNM is described in the following. Here, $\lim_{n \rightarrow \infty} \beta_n = 0$, $\lim_{n \rightarrow \infty} \bar{\beta}_n = 0$ describe the evolutionary trend of VNM. There is $\lim_{n \rightarrow \infty} r_n = 0$, $\lim_{n \rightarrow \infty} \bar{r}_n = 0$.

Proof. It is defined as $P(n) = P\{\vec{X}(n) \cap M = \emptyset\}$.

Based on Bayesian, we have

$$\begin{aligned} P(n+1) &= P\{\vec{X}(n+1) \cap M = \emptyset\} \\ &= P\{\vec{X}(n+1) \cap M = \emptyset / \vec{X}(n) \cap M \neq \emptyset\} \\ &\quad \cdot P\{\vec{X}(n) \cap M \neq \emptyset\} \\ &\quad + P\{\vec{X}(n+1) \cap M = \emptyset / \vec{X}(n) \cap M = \emptyset\} \\ &\quad \cdot P\{\vec{X}(n) \cap M = \emptyset\} \\ &= \alpha_n P\{\vec{X}(n) \cap M \neq \emptyset\} + \beta_n P\{\vec{X}(n) \cap M = \emptyset\}. \end{aligned} \quad (10)$$

Elitist strategy is adopted in VNM, $\alpha_n = 0$. Hence

$$P(n+1) = \beta_n P(n). \quad (11)$$

Then,

$$\begin{aligned} \lim_{n \rightarrow \infty} P(n+1) &= \lim_{n \rightarrow \infty} \beta_n P(n) = \lim_{n \rightarrow \infty} r_n P(0) = 0, \\ \lim_{n \rightarrow \infty} P\{\vec{X}(n) \cap M \neq \emptyset\} &= 1. \end{aligned} \quad (12)$$

Therefore, we have

$$\vec{X}(n) \rightarrow M(P.W.). \quad (13)$$

It means that $\vec{X}(n)$ converges to global optimal solution weakly.

Similarly, it is defined as $P(n) = P\{\vec{X}(n) \cap M^c \neq \emptyset\}$.

By Bayesian formula, we have

$$\begin{aligned} P(n+1) &= P\{\vec{X}(n+1) \cap M^c \neq \emptyset\} \\ &= \bar{\alpha}_n P\{\vec{X}(n) \cap M^c \neq \emptyset\} + \bar{\beta}_n P(n). \end{aligned} \quad (14)$$

Elitist strategy is adopted in VNM, $\lim_{n \rightarrow \infty} \bar{\alpha}_n = 0$. Hence

$$\lim_{n \rightarrow \infty} P(n+1) = \lim_{n \rightarrow \infty} \bar{\beta}_n P(n) = \lim_{n \rightarrow \infty} \bar{r}_n P(0) = 0. \quad (15)$$

Then,

$$\begin{aligned} \lim_{n \rightarrow \infty} P\{\vec{X}(n) \cap M^c = \emptyset\} &= 1, \\ \lim_{n \rightarrow \infty} P\{\vec{X}(n) \in M\} &= 1. \end{aligned} \quad (16)$$

Therefore, we have

$$\vec{X}(n) \rightarrow M(P.S.). \quad (17)$$

It means that $\vec{X}(n)$ converges to global optimal solution strongly. $\square$

4.2. Convergence to Global Optimal. This part focuses on the elitist strategy and proves that the VNM converges to the global optimum from the perspective of transfer matrix.

Theorem 3 (see [35]). $P = \begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$ is a reducible stochastic matrix, where $C : m \times m$ is primitive stochastic matrix and $R, T \neq 0$. Then,

$$\begin{aligned} P^\infty &= \lim_{k \rightarrow \infty} P^k = \lim_{k \rightarrow \infty} \begin{pmatrix} C^k & 0 \\ \sum_{i=0}^{k-1} T^i R C^{k-i} & T^k \end{pmatrix} \\ &= \begin{pmatrix} C^\infty & 0 \\ R^\infty & 0 \end{pmatrix}, \end{aligned} \quad (18)$$

where P^∞ is a stable stochastic matrix with $P^\infty = 1' P^\infty$. $P^\infty = p^0 P^\infty$ is unique regardless of the initial distribution. The matrix P^∞ satisfies that $p_i^\infty > p$ for $1 \leq i \leq m$ and $p_i^\infty = 0$ for $m < i \leq n$.

According to the previous description of VNM, the extended transition matrices for crossover C^+ , mutation M_1^+ , M_2^+ , selection S^+ can be written as block diagonal matrix and upgrade matrix U is lower triangular:

$$\begin{aligned} C^+ &= \begin{pmatrix} C & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C \end{pmatrix}, \quad M_1^+ = \begin{pmatrix} M_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_1 \end{pmatrix}, \\ S^+ &= \begin{pmatrix} S & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S \end{pmatrix}, \quad M_2^+ = \begin{pmatrix} M_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_2 \end{pmatrix}, \\ U &= \begin{pmatrix} U_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ U_{2^{nl}1} & \cdots & U_{2^{nl}2^{nl}} \end{pmatrix}. \end{aligned} \quad (19)$$

C^+ , M_1^+ , S^+ , M_2^+ , and U are with 2^{nl} square matrices. C , M_1 , M_2 , S , and U_{ab} ($1 \leq a, b \leq 2^{nl}$) are all with the size of $n \times n$ (n is the number of individuals and l is the number of individual attributes).

a, b in U_{ab} represents the population's state sequence number (in the order of the populations of the pros and cons from 1 to 2^{nl}). So U is used to represent population's selection process. Each block matrix U_{ab} is a selection of individuals. The details in U_{ab} can be described as: there are some individuals to make $u_{ij} = 1$ established in each row. Firstly, the first individual is compared with all other individuals, $u_{1j} = 1$ if j th individual is optimal (there may be several optima) or $u_{11} = 1$ if no one is better than it. Then,

the second individual is compared with all other individuals except the first individual. The best individual, g th individual, is chosen; set $u_{2g} = 1$ if g th individual is optimal or $u_{22} = 1$ if there is no one better than the second individual. The sorting process continues until all individuals are sorted. To simplify the difficulty of the problem, assume that there is only one global optimal solution set. Then, only U_{11} is a unit matrix, whereas all matrices U_{aa} with $a \geq 2$ are not unit matrices.

In VNM, the populations go through Gauss mutation M_1^+ , crossover C^+ , mutation M_2^+ , selection S^+ , and EP upgrade matrix U . It is worth of noticing that $(\mu + \lambda)$ selection mode is not used in the evolutionary process of VNM and the number of individuals remains unchanged. This means that $S^+ = I$. The transition matrix P^+ for VNM is

$$\begin{aligned}
 P^+ &= M_1^+ C^+ M_2^+ S^+ U \\
 &= \begin{pmatrix} M_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_1 \end{pmatrix} \begin{pmatrix} C & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C \end{pmatrix} \begin{pmatrix} M_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_2 \end{pmatrix} \\
 &\quad \times \begin{pmatrix} S & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & S \end{pmatrix} \begin{pmatrix} U_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ U_{2^{nl}1} & \cdots & U_{2^{nl}2^{nl}} \end{pmatrix} \\
 &= \begin{pmatrix} M_1 C M_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_1 C M_2 \end{pmatrix} \begin{pmatrix} U_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ U_{2^{nl}1} & \cdots & U_{2^{nl}2^{nl}} \end{pmatrix} \\
 &= \begin{pmatrix} M_1 C M_2 U_{11} & 0 & 0 \\ \vdots & \cdots & 0 \\ M_1 C M_2 U_{2^{nl}1} & \cdots & M_1 C M_2 U_{2^{nl}2^{nl}} \end{pmatrix} \\
 &= \begin{pmatrix} P_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ P_{2^{nl}1} & \cdots & P_{2^{nl}2^{nl}} \end{pmatrix}. \tag{20}
 \end{aligned}$$

There is $P_{11} > 0$ in the transition matrix P^+ . The submatrices P_{al} which is with $a \geq 2$ may be gathered in a rectangular matrix $R \neq 0$ so that Theorem 3 can be used to prove that the corresponding VNM converges to the global optimum [36].

5. Experimental Results and Analysis

Computational experiments are carried out to compare the approaches and to evaluate the efficiency of the proposed method. There are two objectives: to minimize the makespan and the mean workload of the instruments. In this section, the performance metric, coverage metric C , is introduced first. There are two experimental instances adopted in this section. They are instances of 30 tasks with 12 instruments and 40 tasks with 12 instruments which are real-world examples taken from a missile system. The instance of 40 tasks with 12 instruments is displayed in Table 1. The instance of 30

TABLE 1: The instance of 40 tasks with 12 instruments.

Task	Scheme	Resource	Time
t_1	w_1^1	r_1, r_7	5
	w_1^2	r_3, r_5	5
	w_1^3	r_6, r_{10}	4
t_2	w_2^1	r_2, r_{11}	5
	w_2^2	r_4, r_9	4
	w_2^3	r_5, r_6	6
	w_2^4	r_3, r_7	4
t_3	w_3^1	r_3	7
	w_3^2	r_{12}	5
t_4	w_4^1	r_9	25
	w_4^2	r_{10}	22
t_5	w_5^1	r_{12}	14
t_6	w_6^1	r_1, r_4	7
	w_6^2	r_3, r_7	8
	w_6^3	r_6, r_8	8
t_7	w_7^1	r_1, r_2	4
	w_7^2	r_3, r_8	2
	w_7^3	r_7, r_{11}	3
t_8	w_8^1	r_1, r_3	5
	w_8^2	r_6, r_{10}	4
	w_8^3	r_7, r_{12}	7
t_9	w_9^1	r_1, r_4	11
	w_9^2	r_7, r_9	13
	w_9^3	r_8, r_{11}	12
t_{10}	w_{10}^1	r_2	9
	w_{10}^2	r_4	10
	w_{10}^3	r_{10}	10
t_{11}	w_{11}^1	r_2, r_7	6
	w_{11}^2	r_3, r_{12}	9
	w_{11}^3	r_8, r_9	8
t_{12}	w_{12}^1	r_2	11
	w_{12}^2	r_5	13
	w_{12}^3	r_{11}	15
t_{13}	w_{13}^1	r_2	4
	w_{13}^2	r_8	5
	w_{13}^3	r_9	7
t_{14}	w_{14}^1	r_3	7
	w_{14}^2	r_{11}	10
	w_{14}^3	r_{12}	8
t_{15}	w_{15}^1	r_{12}	2
t_{16}	w_{16}^1	r_2	9
	w_{16}^2	r_5	7
	w_{16}^3	r_8	6
t_{17}	w_{17}^1	r_1, r_{10}	10
	w_{17}^2	r_5, r_9	12
	w_{17}^3	r_{11}, r_{12}	11
t_{18}	w_{18}^1	r_6	15
t_{19}	w_{19}^1	r_2	8
	w_{19}^2	r_5	7
	w_{19}^3	r_{10}	7
	w_{19}^4	r_{12}	6
	w_{19}^5	r_3	6
t_{20}	w_{20}^1	r_6	4
	w_{20}^2	r_9	5
	w_{20}^3	r_9	5
t_{21}	w_{21}^1	r_1, r_4	2
	w_{21}^2	r_3, r_5	5
	w_{21}^3	r_6, r_8	3

TABLE 1: Continued.

Task	Scheme	Resource	Time
t_{22}	w_{22}^1	r_2	3
	w_{22}^2	r_4	4
	w_{22}^3	r_6	3
	w_{22}^4	r_{10}	4
t_{23}	w_{23}^1	r_3	5
	w_{23}^2	r_{12}	5
t_{24}	w_{24}^1	r_4	14
	w_{24}^2	r_{11}	17
t_{25}	w_{25}^1	r_7	19
t_{26}	w_{26}^1	r_1, r_4	7
	w_{26}^2	r_3, r_7	8
	w_{26}^3	r_6, r_8	10
t_{27}	w_{27}^1	r_1, r_2	2
	w_{27}^2	r_1, r_7	2
	w_{27}^3	r_3, r_8	4
t_{28}	w_{28}^1	r_1, r_3	5
	w_{28}^2	r_4, r_5	4
	w_{28}^3	r_7, r_{12}	2
t_{29}	w_{29}^1	r_1, r_4	11
	w_{29}^2	r_3, r_4	15
	w_{29}^3	r_7, r_8	12
t_{30}	w_{30}^1	r_1	9
	w_{30}^2	r_4	12
	w_{30}^3	r_{12}	10
t_{31}	w_{31}^1	r_2, r_3	6
	w_{31}^2	r_5, r_{11}	8
	w_{31}^3	r_6, r_9	8
t_{32}	w_{32}^1	r_2	11
	w_{32}^2	r_5	13
	w_{32}^3	r_6	17
t_{33}	w_{33}^1	r_2	6
	w_{33}^2	r_6	5
	w_{33}^3	r_{11}	4
t_{34}	w_{34}^1	r_3	7
	w_{34}^2	r_7	8
	w_{34}^3	r_{12}	10
t_{35}	w_{35}^1	r_9	2
t_{36}	w_{36}^1	r_2	9
	w_{36}^2	r_5	7
	w_{36}^3	r_{10}	6
t_{37}	w_{37}^1	r_1, r_2	10
	w_{37}^2	r_7, r_{11}	7
	w_{37}^3	r_5, r_{12}	11
t_{38}	w_{38}^1	r_{10}	15
t_{39}	w_{39}^1	r_4	8
	w_{39}^2	r_6	7
	w_{39}^3	r_9	7
	w_{39}^4	r_{10}	6
t_{40}	w_{40}^1	r_3	6
	w_{40}^2	r_6	5
	w_{40}^3	r_9	5

TABLE 2: Parameters setting.

Population	Generation	CR	F_1	F_2	P
100	250	0.5	1	1	0.05

tasks with 12 instruments is the first 30 tasks in Table 1. The experiment of selection of controlling curve for neighborhood size is shown in Section 5.2. The verification of the improvements of the algorithm is displayed in Section 5.3. In Section 5.3, VNM is compared with MOEA/D. In Section 5.4, the proposed algorithm (VNM) is compared with the variations of CNSGA using real-world TTSP problems. All of the algorithms are executed using 50 independent runs. In all of the experiments, the better performances are denoted in bold. The basic algorithm parameter settings are displayed in Table 2. CR, F_1 , and F_2 are the three control variables for the crossover. p is the mutation probability.

5.1. Performance Metric. For multiobjective optimization, the convergence to the Pareto-optimal set is the most important target to be considered. There are mainly two metrics to evaluate the convergence. One is convergence metric γ , and the other is convergence metric C . The true set of Pareto-optimal solutions is necessary for the calculation of γ . However, the solutions space of TTSP is so large that the true set of Pareto-optimal solutions cannot be obtained by enumeration. The metric C can be used to compare the performances of the two solutions sets obtained by different algorithms. The calculation of C needs only the information of the two solutions sets. Therefore, in this paper the convergence metric C is used to evaluate the performance of the proposed algorithm.

Assume that A and B are two sets of nondominated solutions, and $C(A, B)$ is the ratio of the solutions in B that are dominated by at least one solution in A . Hence,

$$C(A, B) = \frac{|\{x \in B \mid \exists y \in A : y \text{ dominates } x\}|}{|B|}, \quad (21)$$

$C(A, B) = 1$ means that all of the solutions in B are dominated by solutions in A , and $C(A, B) = 0$ means that there is no solution in B dominated by a solution in A . Generally speaking, if $C(A, B) > C(B, A)$, then solution set A is better than solution set B .

5.2. The Selection of Controlling Curve. In this section, three curves are designed and tested to identify the best T controlling curve. $M1$, $M2$, and $M3$, respectively, represent the straight line, monotonic parabolic and nonmonotonic parabolic curves shown in Figure 4. In curve $M2$, the curvature will be 0 at the end of the evolution period. Because of the influence of neighboring updating in MOEA/D, many duplicate solutions will be presented in the final evolution process of MOEA/D. Therefore, Starting Mutation is applied to the beginning of the next iteration to maintain the

FIGURE 5: The boxplot of three curves for 30 * 12 instance.

TABLE 3: Comparison of influence of three curves for 30 * 12 instance.

	Average	Times
$C(M1, M2)$	0.2213	13
$C(M2, M1)$	0.5196	37
$C(M2, M3)$	0.4964	36
$C(M3, M2)$	0.2069	14

population diversity. Tables 3 and 4 show the comparison of the influence of the three curves on the algorithm using two instances. The results show that the monotonic parabolic curve $M2$ has the best performance. This means that the monotonic curve with the fastest rate of change of curvature is the most useful for the algorithm. And the boxplots of three curves for 30 * 12 and 40 * 12 instances in Figures 5 and 6 also give the same conclusion.

5.3. Experiments for Comparisons of VNM and MOEA/D. In order to verify the improvement of VNM, 30 * 12, and 40 * 12 instances are used to test the performance of VNM and MOEA/D. The monotonic parabolic curve $M2$ is selected as the controlling curve in VNM. The neighborhood size in MOEA/D is 20. V and M , respectively, represent VNM and MOEA/D. The results in Tables 5 and 6 show that the concave curve with the fastest rate of change of curvature obtained improvement for VNM. The selected curve renders the size of the neighborhood more suitable than before.

The results of the two independent experiments for comparison of VNM and MOEA/D are shown in Figures 7 and 8 for the 30 * 12 and 40 * 12 instances, respectively. As shown in the figures the solutions obtained by the VNM dominate most of the solutions obtained by MOEA/D. Variable neighborhood and Starting Mutation improve the performance of MOEA/D efficiently.

Figures 9 and 10 are the boxplots for comparison of VNM and MOEA/D. It shows that the data distribution of VNM is superior to MOEA/D. VNM has the better performance

FIGURE 6: The boxplot of three curves for 40 * 12 instance.

TABLE 4: Comparison of influence of three curves for 40 * 12 instance.

	Average	Times
$C(M1, M2)$	0.244178	14
$C(M2, M1)$	0.501508	36
$C(M2, M3)$	0.533806	38
$C(M3, M2)$	0.242146	12

TABLE 5: Comparison of VNM and MOEA/D for 30 * 12 instance.

	Average	Times
$C(V, M)$	0.4845	35
$C(M, V)$	0.2104	15

TABLE 6: Comparison of VNM and MOEA/D for 40 * 12 instance.

	Average	Times
$C(V, M)$	0.5256	40
$C(M, V)$	0.1949	10

because of application of variable neighborhood and Starting Mutation.

5.4. Experiments for Comparisons of VNM and CNSGA. In this section, the VNM is compared with the CNSGA for TTSP. CNSGA is based on NSGA-II. NSGA-II has been successfully applied to job shop scheduling problems [37], reactive power dispatch problems [38], and many other applications. CNSGA has successfully been adopted to solve TTSP [20]. Therefore, a comparison of VNM and CNSGA is carried out to test the performance of the proposed algorithm VNM.

There are two chaotic sequences, logistic map and cat map, and the chaotic sequences can be applied in three positions, population initialization, crossover, and mutation. Therefore, there are six combinations for CNSGA. The nomenclatures for six variants of CNSGA are shown in

FIGURE 7: The comparison of VNM and MOEA/D for 30 * 12 instance.

FIGURE 8: The comparison of VNM and MOEA/D for 40 * 12 instance.

Table 7. Tables 8 and 9 show the comparison of VNM and CNSGA for 30 * 12 and 40 * 12 instances. The VNM is represented by V. All the comparisons between VNM and the variations of CNSGA are based on 50 independent experiments. The average of C metric and better computational times are used for the performance analysis. The results from Tables 8 and 9 show that VNM provides the best performance not only for the average metric C but also in terms of better computational times than CNSGA.

Figures 11 and 12 show the comparisons of VNM and the 6 variations of CNSGA for 30 * 12 instance. The figures show

TABLE 7: Nomenclature for six variants of the CNSGA.

	The logistic map	The cat map
Initial population	L_1	C_1
Crossover operator	L_2	C_2
Mutation operator	L_3	C_3

that the solutions obtained by the VNM dominate most of the solutions obtained by the 6 variations of CNSGA. Therefore, the VNM obtains the best performance.

FIGURE 9: The boxplot of comparison of VNM and MOEA/D for 30 * 12 instance.

FIGURE 10: The boxplot of comparison of VNM and MOEA/D for 40 * 12 instance.

The results of the comparison of VNM and CNSGA for the 40 * 12 instance are shown in Figures 13 and 14. The results also show that the solutions obtained by VNM have higher quality.

In addition, box plots are used to display the performances of the algorithms. The box plots of C metric for VNM and CNSGA with 30 * 12 and 40 * 12 instances are shown from Figures 15, 16, 17, and 18.

From the box plots of C metric, it is clear that the median of VNM is larger than that of the variations of CNSGA in both the 30 * 12 and 40 * 12 instances, and the data distribution of VNM is more reasonable as well. Additionally, the average of VNM is also superior. The results show that VNM demonstrates better performance than CNSGA in solving the multiobjective TTSP. The performance of solutions obtained by VNM is better than that obtained by CNSGA because that the variable neighborhood is adopted in VNM. The span of information exchange in VNM changes following the evolutionary process, but that in CNSGA stays

TABLE 8: Comparison of VNM and six variations of CNSGA for 30 * 12 instance.

	Average	Times
$C(V, L_1)$	0.4206	36
$C(L_1, V)$	0.2200	14
$C(V, L_2)$	0.4077	34
$C(L_2, V)$	0.2648	16
$C(V, L_3)$	0.4182	36
$C(L_3, V)$	0.2248	14
$C(V, C_1)$	0.4638	35
$C(C_1, V)$	0.2210	15
$C(V, C_2)$	0.4602	34
$C(C_2, V)$	0.2288	16
$C(V, C_3)$	0.4128	35
$C(C_3, V)$	0.2525	15

TABLE 9: Comparison of VNM and six variations of CNSGA for 40 * 12 instance.

	Average	Times
$C(V, L_1)$	0.5243	37
$C(L_1, V)$	0.2264	13
$C(V, L_2)$	0.5359	36
$C(L_2, V)$	0.2282	14
$C(V, L_3)$	0.5218	38
$C(L_3, V)$	0.2338	12
$C(V, C_1)$	0.5044	36
$C(C_1, V)$	0.2138	14
$C(V, C_2)$	0.4844	35
$C(C_2, V)$	0.2169	15
$C(V, C_3)$	0.5116	37
$C(C_3, V)$	0.2055	13

the same. The information from the process of evolution helps VNM get better performance.

The variable neighborhood provides a strategy to improve the performance of the algorithm. For different problems with different scales, the controlling curves for the neighborhood size will be different. The Starting Mutation can be also applied to solve other optimization problem in the evolution process. The strategies proposed in this paper can be investigated in other scheduling problem similar to TTSP.

6. Conclusion

How to improve the test efficiency is more and more important in modern industry. TTSP has important application

FIGURE 11: The comparison of VNM and three variations with logistic map for CNSGA for 30 * 12 instance.

FIGURE 13: The comparison of VNM and three variations with logistic map for CNSGA for 40 * 12 instance.

FIGURE 12: The comparison of VNM and three variations with cat map for CNSGA for 30 * 12 instance.

FIGURE 14: The comparison of VNM and three variations with cat map for CNSGA for 40 * 12 instance.

value in modern manufacturing process. TTSP is combinational optimization problem. The final best solutions only account for a rather small subset of the search space. In order to help the solutions avoid being trapped in local optima, this paper proposed a new genetic evolutionary multiobjective optimization algorithm (VNM) to solve the TTSP. The variable neighborhood and Starting Mutation strategies are adopted in VNM to make the crossover span more suitable and improve the diversity of population. Three controlling curves for neighborhood size are studied. The experimental results have shown that the monotonic

parabolic has the best performance. From the experiment conducted for comparison of VNM and MOEA/D, we see that the improved algorithm has made great progress in solving the TTSP problem. And the experiment conducted for comparison of VNM and CNSGA also shows that the improved algorithm is superior to CNSGA in solving TTSP. VNM can also be applied to solve other combinational optimization problems such as FJSP and TSP. Future work will focus on two objectives: the precedence constraint will be added to the TTSP, and information regarding bottleneck tasks will be considered.

FIGURE 15: The boxplot of VNM and three variations with logistic map for CNSGA for 30 * 12 instance.

FIGURE 16: The boxplot of VNM and three variations with cat map for CNSGA for TTSP 30 * 12 instance.

FIGURE 17: The boxplot of VNM and three variations with logistic map for CNSGA for 40 * 12.

FIGURE 18: The boxplot of VNM and three variations with cat map for CNSGA for 40 * 12.

Conflict of Interests

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

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grant no. 61101153 and the National 863 Hi-Tech R and D Plan under Grant 2011AA110101.

References

- [1] A. Radulescu, C. Nicolescu, A. J. C. van Gemund, and P. P. Jonker, "CPR: mixed task and data parallel scheduling for distributed systems," in *Proceeding of 15th Parallel and Distributed Processing Symposium*, pp. 1–9, San Francisco, Calif, USA, April 2001.
- [2] Z. Yin, J. Cui, Y. Yang, and Y. Ma, "Job shop scheduling problem based on DNA computing," *Journal of Systems Engineering and Electronics*, vol. 17, no. 3, pp. 654–659, 2006.
- [3] Y. Zuo, H. Y. Gu, and Y. G. Xi, "Modified bottleneck-based heuristic for large-scale job-shop scheduling problem with a single bottleneck," *Journal of Systems Engineering and Electronics*, vol. 18, no. 3, pp. 556–565, 2007.
- [4] N. Al-Hinai and T. Y. Elmekaw, "Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm," *International Journal of Production Economics*, vol. 132, no. 2, pp. 279–291, 2011.
- [5] J. C. Chen, C.-C. Wu, C.-W. Chen, and K.-H. Chen, "Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10016–10021, 2012.
- [6] L. De Giovanni and F. Pezzella, "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem," *European Journal of Operational Research*, vol. 200, no. 2, pp. 395–408, 2010.
- [7] D. Lei, "Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling," *Applied Soft Computing Journal*, vol. 12, no. 8, pp. 2237–2245, 2012.

- [8] R. Zhang and C. Wu, "A hybrid immune simulated annealing algorithm for the job shop scheduling problem," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 79–89, 2010.
- [9] P. Damodaran and M. C. Véllez-Gallego, "A simulated annealing algorithm to minimize makespan of parallel batch processing machines with unequal job ready times," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1451–1458, 2012.
- [10] K. Li, Y. Shi, S.-L. Yang, and B.-Y. Cheng, "Parallel machine scheduling problem to minimize the makespan with resource dependent processing times," *Applied Soft Computing Journal*, vol. 11, no. 8, pp. 5551–5557, 2011.
- [11] Q. Zhang, H. Manier, and M.-A. Manier, "A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times," *Computers and Operations Research*, vol. 39, no. 7, pp. 1713–1723, 2012.
- [12] W. Teekeng and A. Thammano, "A combination of shuffled frog leaping and fuzzy logic for flexible job-shop scheduling problems," *Procedia Computer Science*, vol. 6, pp. 69–75, 2011.
- [13] G. Zhang, X. Shao, P. Li, and L. Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem," *Computers and Industrial Engineering*, vol. 56, no. 4, pp. 1309–1318, 2009.
- [14] S. H. A. Rahmati and M. Zandieh, "A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 9–12, pp. 1115–1129, 2012.
- [15] X. Y. Shao, W. Q. Liu, Q. Liu, and C. Y. Zhang, "Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem," *The International Journal of Advanced Manufacturing Technology*, vol. 67, no. 9–12, pp. 2885–2901, 2013.
- [16] R. Xia, M. Xiao, J. Cheng, and X. Fu, "Optimizing the multi-UUT parallel test task scheduling based on multi-objective GASA," in *Proceedings of the 8th International Conference on Electronic Measurement and Instruments (ICEMI '07)*, pp. 4839–4844, Xi'an, China, August 2007.
- [17] J. Y. Fang, H. H. Xue, and M. Q. Xiao, "Parallel test tasks scheduling and resources configuration based on GA-ACA," *Journal of Measurement Science and Instrumentation*, vol. 2, no. 4, pp. 321–326, 2011.
- [18] H. Lu, X. Chen, and J. Liu, "Parallel test task scheduling with constraints based on hybrid particle swarm optimization and taboo search," *Chinese Journal of Electronics*, vol. 21, no. 4, pp. 615–618, 2012.
- [19] X. Liang, B. G. Dong, H. Gao, and D. S. Yan, "Parallel test task scheduling of aircraft electrical system based on cost constraint matrix and ant colony algorithm," in *Proceeding of IEEE 10th International Conference on Industrial Informatics*, pp. 178–183, Beijing, Chinese, July 2012.
- [20] H. Lu, R. Y. Niu, J. Liu, and Z. Zhu, "A chaotic non-dominated sorting genetic algorithm for the multi-objective automatic test task scheduling problem," *Applied Soft Computing*, vol. 13, no. 5, pp. 2790–2802, 2013.
- [21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [22] B. Show, V. Mukherjee, and S. P. Ghoshal, "Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm," *Electrical Power and Energy Systems*, vol. 55, pp. 29–40, 2014.
- [23] L. Wang and C. Singh, "Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm," *Electric Power Systems Research*, vol. 77, no. 12, pp. 1654–1664, 2007.
- [24] O. Abedinia, D. Garmarodi, R. Rahbar, and F. Javidzadeh, "Multi-objective environmental/economic dispatch using interactive artificial bee colony algorithm," *Journal of Basic and Applied Scientific Research*, vol. 2, no. 11, pp. 11272–11281, 2012.
- [25] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [26] Y.-Y. Tan, Y.-C. Jiao, H. Li, and X.-K. Wang, "MOEA/D+ uniform design: a new version of MOEA/D for optimization problems with many objectives," *Computers and Operations Research*, 2012.
- [27] C.-M. Chen, Y.-P. Chen, and Q. Zhang, "Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 209–216, Trondheim, Norway, May 2009.
- [28] M. A. Jan and R. A. Khanum, "A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D," *Applied Soft Computing*, vol. 13, no. 1, pp. 128–148, 2012.
- [29] W. Peng, Q. Zhang, and H. Li, "Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem," *Studies in Computational Intelligence*, vol. 171, pp. 309–324, 2009.
- [30] P. A. Carl, *Communication with Automata*, Applied Data Research, Princeton, NJ, USA, 1966.
- [31] T. Zhang, B. Guo, and Y. Tan, "Capacitated stochastic coloured Petri net-based approach for computing two-terminal reliability of multi-state network," *Journal of Systems Engineering and Electronics*, vol. 23, no. 2, pp. 304–313, 2012.
- [32] B. Bollobas, *Graph Theory*, Springer, New York, NY, USA, 1979.
- [33] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic, Boston, Mass, USA, 1999.
- [34] Y.-R. Zhou, H.-Q. Min, X.-Y. Xu, and Y.-X. Li, "Multi-objective evolutionary algorithm and its convergence," *Chinese Journal of Computers*, vol. 27, no. 10, pp. 1415–1421, 2004.
- [35] M. Iosifescu, *Finite Markov Processes and Their Applications*, Dover, 1980.
- [36] Q. Zhang, L. Dong, F. Jiang, and X. J. Zhu, "Convergence of multi-objective evolutionary computation to its pareto optimal set," *Systems Engineering and Electronics*, vol. 22, no. 8, pp. 17–21, 2000.
- [37] E. Moradi, S. M. T. Fatemi Ghomi, and M. Zandieh, "Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7169–7178, 2011.
- [38] S. Jayadevi, S. Baskar, C. K. Babulal, and M. Willjuice Iruthayarajan, "Solving multiobjective optimal reactive power dispatch using modified NSGA-II," *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 2, pp. 219–228, 2011.