

# Research on the Resource-constrained Test task Scheduling Problem of Materiel

Xixiang Chen

Technology Department, Taiyuan Satellite Launching Center, Taiyuan 030027

chen\_xixiang@126.com

**Keywords:** Testability scheme Resource constrained Test task scheduling Multi-objective genetic algorithm Genetic operators

**Abstract.** Resource-constrained test task scheduling problem (RCTSP) is a key problem to testability scheme optimization of materiel. Analyzing the constraints of resources and technologies in testing, a multi-objective optimization model is **founded** in which objectives are testing time and testing cost, and an improved multi-objective genetic algorithm is applied to solve the model. In view of the characteristic of the problem, the chromosome is encoded by way of task-resource dual form, and the rational genetic operators of selection, crossover and mutation are designed to prevent the generation of illegal solutions which can avoid loss of excellent individuals in the parent generation and maintain the diversity among population members. The computation results show that the proposed method is feasible and effective in solving the RCTSP compared with traditional methods.

## Introduction

More and more onerous tasks constantly emergence in testing as the functions and structures of the complexity of huge materiel is being improved. It makes increasing test requirements be a severe challenge. However, available test resources are limited in process of testability scheme optimization. The competitions and the conflicts among the resources will inevitably happen in case of different requests proposed by more than one test tasks at one time[1-3]. Therefore, it is a key problem to schedule the test tasks reasonably under the condition of constrained resources.

In product manufacturing, the scheduling problem has been deeply researched[4-13], and is mainly focused on the following two aspects, firstly, the application of intelligent algorithms improve the efficiency and veracity of optimization relative to the heuristic or exhaustive algorithm. Secondly, as the emergence of new-style problems generating along with practical problems, traditional scheduling models cannot describe the essence of scheduling problems. It is urgently required to found the new models based on the essence of the problem, which widen the research and application space, and receive extensive attention from researchers [14].

When combining complexities of the scheduling and characteristics of the design for testability, an optimization model of resource-constrained test task scheduling problem (RCTSP) is founded in the context of the rocket. In the model, objective functions are minimizing the testing time and testing cost with the constraints of test tasks priorities and limited test resources. Then the multi-objective genetic algorithm (MOGA) is improved to solve the problem. In the end of the paper the conclusions are drawn by an experiment.

## Description and Modeling of RCTSP

In the process of rocket testability scheme optimization, the process of test task scheduling receives the decision-making information of test selections and takes the resource requirements in testing and diagnosing as basic preconditions. It then arranges the test tasks for the limited test resources based on some certain rules so as to perform the rocket testing effectively and efficiently. The test task scheduling plays an important role in utilization the resources reasonably. It can improve the test efficiency, reduce the testing cost and meet the consumer's needs furthest. The decision-making level is a key factor that ensures the test process goes steadily. In Table 1 the test

tasks of some kind of rocket are listed, and there are ten available test resources for the rocket. The occupation relationships between its test tasks and resources are shown in Table 2.

The essence of RCTSP is locating and dispatching the appropriate test tasks for the resources in terms of minimizing time and cost. The background and variety of constraints between test tasks and resources endow the scheduling problem with more additional characteristics, which increase the complexity of the scheduling. In the section the problem is analyzed and a multi-objective optimization model for RCTSP will be founded by considering the priorities of test tasks and the occupation situation in their resources.

Table 1 The system-level test tasks of the rocket

No.	Candidate Test task	No.	Candidate Test task	No.	Candidate Test task
$MT_1$	DC power supply for inertial measuring units	$MT_7$	RS422 interface of remote transponder	$MT_1$	Rudder instruction
$MT_2$	DC power supply for fuze	$MT_8$	RS422 interface of fuze	$MT_1$	Zero-effort current of actuator feedback
$MT_3$	DC power supply for seeker	$MT_9$	RS422 interface of seeker	$MT_1$	Remote initiation presetting of remote transponder
$MT_4$	DC power supply before launch	$MT_{10}$	Parallel interface of inertial measuring units	$MT_1$	Selected gate of echo signal
$MT_5$	DC power supply away from shelf	$MT_1$	Communication interface between rocket and ground	$MT_1$	Implement voltage
$MT_6$	DC power supply for actuator	$MT_1$	Remote initiation instruction	$MT_1$	Transfer signal

Table 2 The occupation relationships of the rocket test tasks and test resources

Serial Number of Tasks	Available Resources									
	BIT for computer on rocket	BIT for rudder	BIT for remote transponder	BIT for fuze	Digital multimeter	Synthetic analyzer	Digitizer	Oscilloscope	Analyzer for DC power	Ground computer
$MT_1$	—	—	—	—	5(8)*	3(7)	8(8)	9(3)	5(10)	—
$MT_2$	—	—	—	—	8(12)	10(6)	4(9)	11(2)	4(15)	—
$MT_3$	—	—	—	—	6(4)	9(6)	5(5)	8(10)	3(11)	—
$MT_4$	—	—	—	—	8(9)	4(4)	5(8)	8(5)	4(7)	—
$MT_5$	—	—	—	—	6(6)	3(9)	10(11)	7(13)	8(12)	—
$MT_6$	—	—	—	—	15(13)	5(8)	14(14)	9(5)	6(13)	—
$MT_7$	7(7)	—	—	—	—	—	—	—	—	—
$MT_8$	9(9)	—	—	—	—	—	—	—	—	—
$MT_9$	7(8)	—	—	—	—	—	—	—	—	—
$MT_{10}$	3(9)	—	—	—	—	—	—	—	—	—
$MT_{11}$	—	—	—	—	—	—	—	—	—	2(4)
$MT_{12}$	—	—	4(5)	—	—	—	—	—	—	—
$MT_{13}$	—	6(4)	—	—	—	—	—	—	—	—
$MT_{14}$	—	4(7)	—	—	—	—	—	—	—	—
$MT_{15}$	—	—	8(2)	—	—	—	—	—	—	—
$MT_{16}$	—	—	5(2)	—	—	—	—	—	—	—
$MT_{17}$	—	—	—	8(4)	—	—	—	—	—	—
$MT_{18}$	—	—	—	—	—	3(3)	5(10)	2(12)	—	—

Note\*: the values in table indicate the implement time (or cost) of test tasks tested by the corresponding resources.

### Description of the Problem

In general, the test task scheduling is not only constrained by test tasks, but constrained by test personnel and test resources. The RCTSP can be described as follows, suppose that the system has  $N$  test tasks  $MT$  implemented by  $M$  available resources  $R$ . For the example of Table 2, we have  $N=18$ ,  $M=10$ , and the information of testing time and testing cost of a certain rocket is presented. The purpose is to settle the conflicts among the limited resources in the process of testing so that time and cost can be minimized simultaneously under the condition of satisfying priorities of test tasks. Now some symbols are defined as follows before the optimization model is founded.

- I The rocket has  $N$  tasks to be tested  $MT=\{MT_1, MT_2, \dots, MT_N\}$ .

- ┃ The set of test resources  $\mathbf{R}$ , there are  $M$  resources available,  $\mathbf{R}=\{R_1, R_2, \dots, R_M\}$ .
- ┃ The testing cost matrix, whose element  $C_i^k$  denotes the cost spent during testing the  $i$ th task by means of  $k$ th resource.
- ┃ The testing time matrix, whose element  $T_i^k$  denotes the time spent during testing the  $i$ th task by means of  $k$ th resource.
- ┃ The priority constraints among test tasks  $P_{ij}^{kg}$ , if the  $i$ th task tested by the  $k$ th resource is carried out before the  $j$ th task tested by the  $g$ th resource,  $P_{ij}^{kg} = 1$ , otherwise  $P_{ij}^{kg} = 0$ .
- ┃ The exclusive matrix  $\mathbf{E}$ , the value of whose element  $E_{ij}$  equals 1, i.e.  $E_{ij}=1$ , if  $i$ th task and the  $j$ th task can not be carried out simultaneously, otherwise  $E_{ij}=0$ .
- ┃ The task dispatching matrix  $\mathbf{X}$ , if the  $i$ th task is implemented by the  $k$ th test resource,  $X_i^k = 1$ , otherwise,  $X_i^k = 0$ .
- ┃ The beginning time  $ST_i^k$ , which denotes the beginning time of the  $i$ th task carried out by the  $k$ th resource.
- ┃ The finishing time  $ET_i^k$ , which denotes the finishing time of the  $i$ th task carried out by the  $k$ th resource.

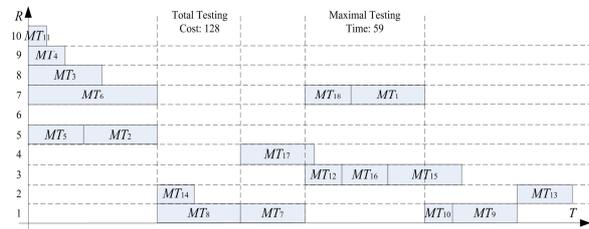


Fig 1 A scheduling of the instance shown in Table 2

Based on the descriptions above, the  $N$ -dimension vector  $\mathbf{S} = (ST_i^k)_{N \times 1}$  ( $\forall i = 1, 2L, N$  and  $X_i^k = 1$ ) is named a scheduling or a solution of RCTSP. And a scheduling is called a feasible one if the following supposes are satisfied. A solution is showed in Fig. 1.

- ┃ All tasks that do not have anterior tasks can begin at time 0, and arrived time of the first task carried out is also at time 0.
- ┃ Sequential constraint of test tasks—the tasks that have sequential constraint relations have to be carried out sequentially, i.e.  $ST_i^k - ST_j^g \geq T_j^g$ , under the condition  $X_i^k = X_j^g = 1$  and  $P_{ij}^{kg} = 1$ .
- ┃ The constraints of test resources occupation—a task have to be carried out by the resource after its former task carried out by the same one is finished, i.e.  $ST_i^k - ST_j^k \geq T_j^k$  under the condition  $X_i^k = X_j^k = 1$  and  $P_{ij}^{kk} = 1$ .
- ┃ It is not allowed interruption when a task is beginning until it is finished, i.e.  $ET_i^k - ST_i^k = T_i^k$ .

### Foundation of Scheduling Model

In a word, RCTSP can be described as a 4-cell model, i.e.  $\{\mathbf{MT}, \mathbf{R}, \mathbf{Cons}, \mathbf{O}\}$ , where  $\mathbf{MT}$ ,  $\mathbf{R}$ , and  $\mathbf{Cons}$  denote test tasks, test resources and constraints respectively.  $\mathbf{O}$  consists of two objectives.

(1) Minimizing maximal testing time. The testing time is indicated the finishing time of the last task.

$$O_1 = \min(\max_i(\max_k(ET_i^k))) \quad (1)$$

(2) Minimizing total testing cost. The total testing cost is indicated the spending cost of all test tasks.

$$O_2 = \min(\sum_{i,k} C_i^k X_i^k) \quad (2)$$

It can be induced that RCTSP is one of NP-hard problems, and also a multi-objective optimization problem(MOP).

## Test Task Scheduling Based-on Multi-objective Genetic Algorithm

For multi-objective optimization problems (MOPs), optimization of an objective is usually at the cost of degradations of other objectives due to the restrictions between the objectives, and because of their different properties the excellence of solutions can not be evaluated objectively. A group of non-dominated solutions are achieved consequently. While The traditional methods, such as the weighting method and constraint method, transforming the MOP to the single-objective problem, not only destroy the original essence, but also introduce the difficulties in weights computing, shape sensitivity to non-dominated front, and trapping into local optimization etc.

In view of above reasons, the paper presents the flow of an improved multi-objective GA and then designs the corresponding genetic operators so as to suit for the RCTSP.

### Idea of Improved MOGA

Improved non-dominated sorting genetic algorithm (NSGA-II)[16,17] is a popular multi-objective genetic algorithm proposed by Srinivas and Deb in 2000. The algorithm overcomes the inherent defects existed in its predecessor NSGA[15]. Based on the algorithms an improved MOGA is present in the following section. It can enhance the searching efficiency through a more reasonable assignment of the fitness to every population member, and simultaneity reduces the computational complexity by designing the new GA Operators.

#### Procedures of Fitness Assignment in Improved MOGA

In the improved MOGA there are two steps for fitness assignments. Firstly, a fast non-dominated sorting approach (*Fast\_Nondominated\_Sort*) is applied to obtain the non-dominated level of each individual via compute its objective functions, i.e. testing time and testing cost. Individuals having the same non-dominated level are assigned a same non-dominated value and belong to one non-dominated front. In this way, the whole population can be partitioned into different non-dominated fronts according to their non-dominated values. Thereafter, multiple objective functions are transformed to a single value, and then the excellence ranks of all individuals are marked preliminarily based on their non-dominated values. Secondly, for the individuals belonging to the same non-dominated front, density estimation approach (*Density\_Estimation*) is applied to further distinguish the excellence of individuals form each other. The approach computes the crowding-distance between two adjacent individuals of a certain individual according to their objective values, and then the distance values with respect to each objective are summed as the criterion to evaluate the intensive degree of current individual. Every individual has a crowding-distance, and the individual with bigger distance means that the distributing of the population around the current individual is sparser. Herein, two objective functions are transformed to a comparable value ultimately, and the assignment of the fitness is completed.

In addition, a compare operator (*Compare*) is defined in the improved MOGA for comparing the priority between two individuals. The criterion is as follows, for the individuals in different non-dominated front, we prefer the individual with lower non-dominated level. If two individuals belong to the same front, the individual with the smaller density is better. In this way, the diversity of the population is preserved when the optimal solutions are searched repeatedly.

#### Algorithm flow of Improved MOGA

In order to avoid loss of excellent individuals in the parent generation and maintain the diversity among population members, the elitist strategy is introduced to the improved MOGA. In the strategy, when combining the parent and offspring populations, all of the individuals are sorted via non-dominated sorting procedure, and then the genetic operations are executed. The detailed steps are listed below.

Step0: Initializing iteration number or stopping condition.

Step1: Initializing the population  $P_0$  (with  $L$  individuals) stochastically, which is sorted based on non-dominated levels. Then operators such as the tournament selection of size 2, crossover and mutation are applied to  $P_0$ , and the offspring population  $Q_0$  is obtained.

Step2: Setting  $t=0$ .

Step3: Combining  $P_t$  and  $Q_t$  into a new population  $V_t$ , i.e.  $V_t = P_t \cup Q_t$ .

Step4: Applying the non-dominated sorting approach to  $V_t$  and computing  $(PF_1, PF_2, \dots) = \text{Fast\_Nondominated\_Sort}(V_t)$ , thereinto  $PF_i$  denotes the  $i$ th non-dominated front.

Step5: Setting  $i=1$ .

Step6: Applying density estimation approach to  $PF_i$ , i.e.  $\text{Density\_Estimation}(PF_i)$ , and incorporate it into the population  $P_{t+1}$ , i.e.  $P_{t+1} = P_{t+1} \cup PF_i$ .

Step7:  $i=i+1$ . If the size of  $P_{t+1}$  is less than  $L$ , return Step6, or else, continue.

Step8: Choosing first  $L$  individuals of  $P_{t+1}$  to form a new population  $P_{t+1}$ :  $P_{t+1} = P_{t+1}[1 : L]$ .

Step9: The genetic operators such as tournament selection of size 2, crossover and mutation are applied to  $P_{t+1}$  and a new offspring population  $Q_{t+1}$  is generated.

Step10:  $t=t+1$ . If the appointed stopping condition in Step0 is not satisfied, return Step3, or else, stop and export optimal individuals.

## Design of GA Operators

### Encoding and Decoding of Chromosomes

The encoding or decoding is the transformation procedure between chromosomes and solutions, which is the primary process of GA[18]. For the test task scheduling model described in section 2.2, the chromosome consists of two parts.

$$\mathbf{XCode} = \begin{pmatrix} \mathbf{MTCODE} \\ \mathbf{RCODE} \end{pmatrix} = \begin{pmatrix} \mathbf{MTCODE}_1, L, \mathbf{MTCODE}_p, L, \mathbf{MTCODE}_L \\ \mathbf{RCODE}_1, L, \mathbf{RCODE}_q, L, \mathbf{RCODE}_L \end{pmatrix}$$

The first part is **MTCODE** based on test tasks. It adopts the natural numbers to the chromosome of size  $N$  by encoding the test tasks in order, where the  $p$ th value  $\mathbf{MTCODE}_p (1 \leq \mathbf{MTCODE}_p \leq N)$  from left to right denotes the task serial number and exists only once in **MTCODE**. The priority constraints have to be satisfied in task encoding.

The second part is **RCODE** based on test resources. It also adopts the natural numbers to the chromosome of size  $N$  and is according with the tasks coding **MTCODE**, where the  $q$ th  $(1 \leq q \leq N)$  value  $\mathbf{RCODE}_q$  from left to right denotes the corresponding resource **RCODE** <sub>$q$</sub>  used to implement the test task **MTCODE** <sub>$q$</sub> .

During decoding the resource assigned to each task is obtained first, and then the tasks begin to carry out in order at possible earliest time in their respective resources so that the feasible scheduling solutions can be generated.

### Selection Operator

The tournament selection of size 2[19] is introduced in the selection procedure, i.e. two individuals chosen randomly from the parent population is compared based on *Compare*-operator and the better one is selected. If the non-dominated level of two individual are equal, the one with the smaller density is selected as the better one by *Density\_Estimation*.

### Crossover Operator

The procedure of crossover is complex because the priorities existed among the test tasks must be preserved in crossover. In the procedure, two parts in chromosome are taken as a whole to crossover. A task is selected randomly, and the locations of the task in two parent chromosomes under crossover are obtained, then the former position is taken as the crossover point. For example, the task  $MT_{15}$  is selected, whose position 8 in  $\mathbf{XCode}_1$  is smaller than that position 10 in  $\mathbf{XCode}_2$ , therefore the position 8 in  $\mathbf{XCode}_1$  is the crossover point, and then the following operations are executed.

$$\mathbf{XCode}_1 = \begin{pmatrix} 6 & 3 & 11 & 7 & 16 & 14 & 5 & \boxed{15}^8 & 1 & 2 & 8 & 17 & 4 & 18 & 9 & 13 & 10 & 12 \\ 8 & 9 & 10 & 1 & 3 & 2 & 8 & \boxed{3} & 8 & 6 & 1 & 4 & 8 & 6 & 1 & 2 & 1 & 3 \end{pmatrix}$$

$$\mathbf{XCode}_2 = \begin{pmatrix} 2 & 8 & 4 & 5 & 3 & 6 & 1 & 7 & 18 & \boxed{15}^{10} & 16 & 12 & 11 & 17 & 14 & 9 & 13 & 10 \\ 6 & 1 & 5 & 5 & 9 & 9 & 7 & 1 & 6 & \boxed{3} & 3 & 3 & 10 & 4 & 2 & 1 & 2 & 1 \end{pmatrix}$$

①The genes {6, 3, 7, 5} before the position 8 in parent chromosome  $XCode_1$  that is the same as the genes before the position 8 in parent chromosome  $XCode_2$  are exchanged to identical positions of  $XCode_2$  as the genes of offspring chromosome  $XChild_2$ , and these positions in  $XCode_1$  become blank. ②The same genes {5, 3, 6, 7} in  $XCode_2$  are put into the blank positions of  $XCode_1$  in their initial order as the genes of offspring chromosome  $XChild_1$ , and other genes {11, 16, 14} in  $XCode_1$  is unchangeable as the left genes of  $XChild_1$ . ③The left genes {2, 8, 4, 1, 18} in  $XCode_2$  are taken as the other genes of  $XChild_2$  in their initial order, the procedures are shown below.

$$\begin{aligned}
 XCode_1 &= \begin{pmatrix} 6 & 3 & 11 & 7 & 16 & 14 & 5 & \boxed{15} & 1 & 2 & 8 & 17 & 4 & 18 & 9 & 13 & 10 & 12 \\ 8 & 9 & 10 & 1 & 3 & 2 & 8 & \boxed{3} & 8 & 6 & 1 & 4 & 8 & 6 & 1 & 2 & 1 & 3 \end{pmatrix} \\
 XCode_2 &= \begin{pmatrix} 2 & 8 & 4 & 5 & 3 & 6 & 1 & 7 & 18 & \boxed{15} & 16 & 12 & 11 & 17 & 14 & 9 & 13 & 10 \\ 6 & 1 & 5 & 5 & 9 & 9 & 7 & 1 & 6 & \boxed{3} & 3 & 3 & 10 & 4 & 2 & 1 & 2 & 1 \end{pmatrix}
 \end{aligned}$$

The offspring chromosomes are as follows after carrying out the crossover.

$$\begin{aligned}
 XChild_1 &= \begin{pmatrix} 5 & 3 & 11 & 6 & 16 & 14 & 7 & \boxed{15} & 1 & 2 & 8 & 17 & 4 & 18 & 9 & 13 & 10 & 12 \\ 5 & 9 & 10 & 9 & 3 & 2 & 1 & \boxed{3} & 8 & 6 & 1 & 4 & 8 & 6 & 1 & 2 & 1 & 3 \end{pmatrix} \\
 XChild_2 &= \begin{pmatrix} 6 & 3 & 4 & 7 & 2 & 8 & 5 & 1 & 18 & \boxed{15} & 16 & 12 & 11 & 17 & 14 & 9 & 13 & 10 \\ 8 & 9 & 5 & 1 & 6 & 1 & 8 & 7 & 6 & \boxed{3} & 3 & 3 & 10 & 4 & 2 & 1 & 2 & 1 \end{pmatrix}
 \end{aligned}$$

It is noted that  $XChild_2$  is not satisfied the priorities among tasks by any possibility when performing the step ③. There is the need for small adjustments of some genes in such an individual.

**Mutation Operator**

During mutation, the two parts of the chromosome are taken as a whole to deal with. For the problem in Table 2, suppose the sequence of tasks is  $MT_1 \rightarrow MT_{10}$ , and perform mutation on the following chromosome  $XCode$ .

$$XCode = \begin{pmatrix} 6 & 3 & 11 & 7 & 16 & 14 & 5 & 15 & \boxed{1} & 2 & 8 & 17 & 4 & 18 & 9 & 13 & 10 & 12 \\ 8 & 9 & 10 & 1 & 3 & 2 & 8 & 3 & \boxed{8} & 6 & 1 & 4 & 8 & 6 & 1 & 2 & 1 & 3 \end{pmatrix}$$

There are two steps in mutation in order to preserving the priorities of tasks. Firstly, a task is randomly selected from  $XCode$  as a mutation point. For example, the task  $MT_1$  of 9th position in  $XCode$  is selected, and whose successor tasks { $MT_{10}$ } are obtained. Then the gene  $MT_1$  will be inserted to the arbitrary position before its successor tasks, as is shown in  $XCode'$ .

$$XCode' = \begin{pmatrix} 6 & 3 & 11 & 7 & 16 & 14 & 5 & 15 & 2 & 8 & 17 & 4 & \boxed{1} & 18 & 9 & 13 & 10 & 12 \\ 8 & 9 & 10 & 1 & 3 & 2 & 8 & 3 & 6 & 1 & 4 & 8 & \boxed{8} & 6 & 1 & 2 & 1 & 3 \end{pmatrix}$$

Secondly, an arbitrary resource  $R_5$  from available resources of the task  $MT_1$  is selected to substitute for current resource  $R_8$  and the final offspring chromosome  $XChild$  is generated.

$$XChild = \begin{pmatrix} 6 & 3 & 11 & 7 & 16 & 14 & 5 & 15 & 2 & 8 & 17 & 4 & \boxed{1} & 18 & 9 & 13 & 10 & 12 \\ 8 & 9 & 10 & 1 & 3 & 2 & 8 & 3 & 6 & 1 & 4 & 8 & \boxed{5} & 6 & 1 & 2 & 1 & 3 \end{pmatrix}$$

**Experimental Results**

In order to prove the validity of the proposed approach, a comparison is drawn between the approach and the weighting method based on the example in Table 2.

Suppose the constraint matrix arising from priorities is  $P = (p_{ij})_{18 \times 18}$ , in which the sequences are  $MT_2 \rightarrow MT_8$ ,  $MT_3 \rightarrow MT_9$ ,  $MT_1 \rightarrow MT_{10}$ ,  $MT_7 \rightarrow MT_{12}$ ,  $MT_3 \rightarrow MT_{13}$ ,  $MT_6 \rightarrow MT_{13}$ ,  $MT_9 \rightarrow MT_{13}$ ,  $MT_6 \rightarrow MT_{14}$ ,  $MT_7 \rightarrow MT_{15}$ ,  $MT_7 \rightarrow MT_{16}$ ,  $MT_2 \rightarrow MT_{17}$ ,  $MT_8 \rightarrow MT_{17}$ ,  $MT_7 \rightarrow MT_{18}$ , i.e.  $p_{2,8}=1$ ,  $p_{3,9}=1$ ,

$p_{1,10}=1, p_{7,12}=1, p_{3,13}=1, p_{6,13}=1, p_{9,13}=1, p_{6,14}=1, p_{7,15}=1, p_{7,16}=1, p_{2,17}=1, p_{8,17}=1, p_{7,18}=1$ , and other elements of  $P$  equal 0. Then the multi-objective optimization model based on testing time and testing cost can be founded as shown in section 2.2.

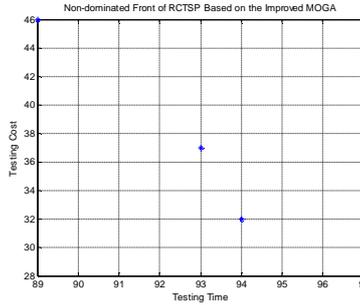


Fig 2 Non-dominated front of RCTSP based on the improved MOGA

When computing, the parameters are set as follows, the size of the population is 50, the iteration number is 100, and the probabilities of crossover and mutation are 0.9 and 0.1 respectively.

The algorithm presented in section 3 is then applied to solve the model. A group of non-dominated solutions are achieved and its non-dominated front is shown in Fig. 2, and  $X\_Best^N$  below is one of the non-dominated solutions.

$$X\_Best^N = \begin{pmatrix} 5 & 1 & 7 & 2 & 11 & 6 & 14 & 15 & 8 & 16 & 3 & 17 & 4 & 18 & 9 & 13 & 10 & 12 \\ 5 & 8 & 1 & 7 & 10 & 8 & 2 & 3 & 1 & 3 & 7 & 4 & 8 & 6 & 1 & 2 & 1 & 3 \end{pmatrix}$$

The optimal solution  $X\_Best^W$  is shown as follows when applied the weighting method.

$$X\_Best^W = \begin{pmatrix} 3 & 1 & 4 & 7 & 16 & 15 & 9 & 18 & 12 & 2 & 8 & 17 & 6 & 14 & 10 & 5 & 13 & 11 \\ 5 & 6 & 8 & 1 & 3 & 3 & 1 & 7 & 3 & 6 & 1 & 4 & 8 & 2 & 1 & 6 & 2 & 10 \end{pmatrix}$$

For the problem in Table 2, the results computed respectively by the algorithm proposed in the paper and the weighting method are shown in Fig.3. Then the comparisons between two methods are shown in Table 3 after the multi-computation.

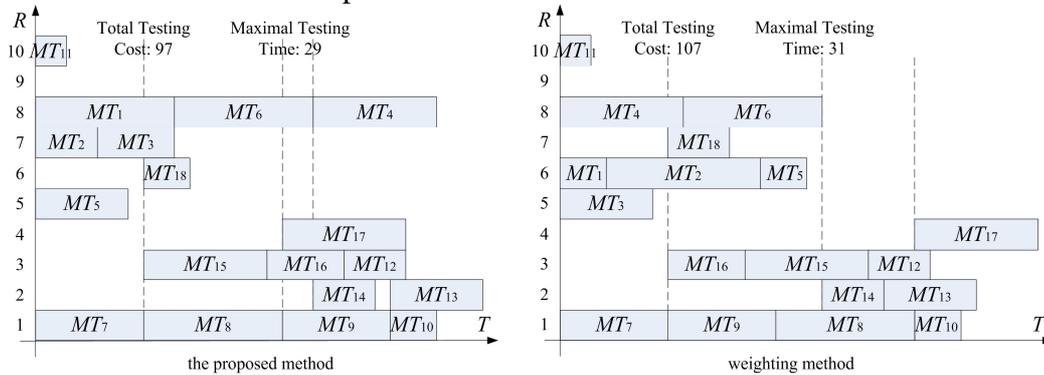


Figure 3 A non-dominated solution of RCTSP based on the improved MOGA compared with that based on the weighting method

Table 3 Comparisons between the two methods

Objectives	Optimal results		Average results	
	Weighting method	Presented method	Weighting method	Presented method
maximal testing time	31	29	36	31
total testing time	107	97	99.5	93.25

## Conclusions

(1) Kinds of constraints of test resources and two important objectives of the most attention are investigated, based on which an optimization model for RCTSP is founded.

(2) In order to avoid loss of excellent individuals in the parent generation and maintain the diversity among population members, the procedure of fitness dispatching is expounded, and the detailed algorithm flow of the improved MOGA is presented.

(3) Aim at the particularities of RCTSP, the chromosome is encoded by way of task-resource dual form. The chromosome indicates the structure of scheduling solution intuitively and easy to decode. Whereafter, the genetic operators of crossover and mutation are newly designed in detail, under which the offspring individuals is still the feasible for the problem.

(4) At the end of the paper, it is proved by the experiment that the approach proposed is feasible and effective. When compared the algorithm with the traditional method, it is shown that the traditional method transforms the multi-objective problem to the single-objective problem, which is divorced from the essence of MOPs and runs into the local optimization. However, the algorithm in the paper can obtain the non-dominated solution as much as possible, and can achieve the optimal solution at the greater opportunity.

## References

- [1] Navy-EC . MIL-STD-2165 . Testability Program For Electronic Systems And Equipments[S]. Washington D.C : Department of Defense(1985).
- [2] GJB 2547-95. Testability program for materiel[S]. Beijing: The Industrial Committee of Defense Science and Technology, Oct(1995).
- [3] CHEN Xixiang, QIU Jing, LIU Guanjun. Research on Technologies of optimal testability concept for materiel system[J]. China Mechanical Engineering, 2010, 21(2),p.141-145.
- [4] JOHNSON S.M. Optimal two and three stage production schedules with setup times included[J]. Naval Research Logistics Quarterly, 1954, 1(1),p.61-68.
- [5] Taubert W H. A search decision rule for the aggregate scheduling problem[J]. Management Science, 2001, 74,p.33-48.
- [6] PAN Quanke. Research on the multi-objective job shop scheduling optimization of intelligent manufacturing systems[D]. Nanjing: Nanjing University of Aeronautics and Astronautics(2003).
- [7] WANG Ling. Shop scheduling with genetic algorithms[M]. Beijing: Tsinghua University Press(2003).
- [8] GONCALVES J F, MENDES J J M, RESENDE M G C. A hybrid genetic algorithm for the job shop scheduling problem[J]. European Journal of Operational Research, 2005, 167(1),p.77-95.
- [9] WANG Hong. Research on the algorithms of resource-constrained project scheduling problem[D]. Tianjin: Tianjin University(2005).
- [10] GUO Dongfen, LI Tieke. Constraint-based algorithm for job shop scheduling[J]. Computer Integrated Manufacturing Systems, 2007, 13(1),p.117-124.
- [11] CHANG Guijuan. Research on the particle swarm optimization based job-shop scheduling[D]. Qingdao: Qingdao University(2008).
- [12] ZHANG Guohui, GAO Liang, LI Peigen, ZHANG Chaoyong. Improved genetic algorithm for the flexible job-shop scheduling problem[J]. Journal of Mechanical Engineering, 2009, 45(7),p.145-151.
- [13] WU Jinghua, TANG Wencheng, XU Hongxiang, BEN Kecun. Collaborative scheduling algorithm for flexible design job[J]. Journal of Mechanical Engineering, 2009, 45(10),p.228-234.
- [14] ZHAO Chuanli, TANG Hengyong. Introduction of Scheduling[M]. Beijing: Science Press(2002).

- [15]SRINIVAS N, DEB K. Multiobjective optimization using nondominated sorting in genetic algorithms[R]. Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India(1993).
- [16]DEB K, AGRAWAL S, PRATAP A, MEYARIVAN T. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization : NSGA-II[C]. Proceedings of the Parallel Problem Solving from Nature VI Conference(2000).
- [17]DEB K, AGRAWAL S, PRATAP A, MEYARIVAN T. A fast and elitist multi-objective genetic algorithm: NSGAIJ[J]. IEEE Transaction on Evolutionary Computation, 2002(6),p.182-197.
- [18]ZHOU ming, SUN Shudong. The theory of genetic algorithm and its application[D]. Beijing: National Defense Industry Press(1999).
- [19]PAN Zhengjun, KANG Lishang, CHEN Liuping. Evolution Computing[M]. Beijing: Tsinghua University Press(1998).