Multi-objective Flexible Scheduling Optimization Scheme base on Improved DNA Genetic Algorithm

NIE Shuzhi
Department of Electronics and Information Technology, Jiangmen Polytechnic, Jiangmen, China
Email: sznie778@163.com

ZHONG Yanhua
Department of Electronics and Information Technology, Jiangmen Polytechnic, Jiangmen, China
Email:zhflowers@163.com

***Abstract*—In this paper, established a mathematical model for multi-objective flexible scheduling problems, combined Pareto non-dominated sorting method, put forward a hybrid genetic algorithm based on improved DNA computation. To ensure the diversity of the optimal solution sets, designed RNA quaternary encoder mode and genetic operator based on improved DNA computation, adopted sub-area crossover and dynamic mutation, imposed on manipulation of the molecular level. Through simulation, tested the performance of the designed algorithm, compared it with the standard genetic algorithm test results. Simulation results showed the proposed algorithm can provide an optimum searching, owned better seeking abilities; the obtained scheduling results were fairly reasonable. This algorithm can effectively solve the multi-objective flexible scheduling optimization problems.**

Index Terms*—DNA computation, RNA genetic operators, Genetic Algorithm, Pareto sorting, Multi-objective flexible scheduling

I. INTRODUCTION

Most of the engineering and scientific problems are the multi-objective optimization problems, there are multiple conflicting with objectives, how to obtain the optimal solutions of these problems. This has been the focus of academic and engineering problems. It is different from the single objective optimization problem, in most cases. The essence of multi-objective optimization is coordination and compromise handling between the various objectives. All the functions as far as possible achieve the optimal values.

Multi-objective flexible scheduling is one of the representative problems of uncertain polynomial hard problems, which seek optimum solutions of multi-objects. Now there are many answers for the flexible scheduling actuality. This study is of practical and theoretical significance. It has a role in guiding production and practice [1, 2].

In recent years, some intelligent calculation methods such as genetic algorithm, neural networks etc... They are being successively used to solve scheduling problems. These optimization methods have made certain application effects on scheduling problems [3, 4]. It has large latent abilities for solving the complex optimization, which has been successfully applied to some industrial projects. Genetic algorithms have attracted wide attention and concern. It was considered to be a very worthwhile research method of intelligent optimization, particularly in the areas of application, development, and production, planning and scheduling [5-8].

Computer science and biological science research in the marriage of information have become a popular frontier. Genetic algorithms, artificial neural networks and ant colony algorithms are all get inspiration from the evolution of biological to explore and solve complex problems. On the other hand, with the successful implementation of the human genome plan, and the beginning of post genome era, the emergence of biological molecular data brings vast amounts data. How to dig and discover the meaning of these data, reveal the mystery of life has become scientists facing a new challenge. Some people predicted that the twenty-first century science will be the main battlefield of life sciences. In this context, Dr. Adleman's article published in Science, "Molecular Computation of Solutions to Combinatorial Problems" marked a new research field of DNA computation was born [9-11].

DNA computation and genetic algorithm exist certain similarities in ideological, DNA computation is a method of molecular biology information coding, used the double-helix structure of DNA and the rule of complementary base pairing [10, 11]. DNA is an important material, carried ample genetic information. It can promote genetic algorithm simulate biological rule, improve the performance of genetic algorithm. Genetic algorithm can be a breakthrough in the limits of DNA computation [12, 13]. Scholars have put forward some algorithms based on DNA computation to solve various complex problems [11-13]. Researchers have divided the DNA molecular sequence into two classes: neutral and harmful. They pointed out that genetic manipulation in

doi:10.4304/jcp.7.8.1982-1989

molecular sequences can cause different evolutionary results. The same DNA sequence exists hot spots and cold spots in different locations; the mutation probability of the base in the cold spots is far less than the base in hot spots [14, 15]. NIU and GU have used DNA evolutionary algorithm to solve flow shop scheduling problems [16-18].

In this paper, uses Pareto non-dominated sorting method, utilizes RNA computation base on improved DNA computation, and exerts the genetic operation on a molecular level to achieve better results.

II. DESCRIPTION OF THE PROBLEM

Flexible job-shop scheduling problems reduce the machine constraints, expand the search scope of feasible solutions, increased the problem's complexity. It is close to realistic simulation of the production environment than traditional job shop scheduling problems. Therefore, obtain the optimal solutions of research problems by various evolutionary methods in engineering. It has become an important research subject in CIMS (Computer Integrated Manufacturing Systems, CIMS) fields, has important theoretical and applied significance [19-21].

Flexible scheduling problems mean to give each work-piece optional path solutions, to find a workable scheduling for each group of a work-piece, to optimize some performance indexes. Details are shown as follows: to process N work-pieces on M machine tools, each work-piece J_i is made up by n_i processes. Among these n_i processes, exists a restrictive order relation in technology, a work-piece can be processed by many machine tools out of M machine tools. The processing time differs from the performance of these machine tools [22, 23]. The manufacturing environment is different, made the following assumption: each machine tool can process only one work-piece at one time. The processing time is pre-determined. All machine tools are perfect and workable at the time of t =0. All work-pieces process at the time of t =0. The machine damage is ignored [24, 25].

The hybrid scheduling objectives of flexible scheduling are as follows: as much as possible to shorten general time of delivery and improve customers' satisfaction. Decrease the load of machine tools and increase the use efficiency of resources available. Then a production period f_1, the total deferred time f_2, the largest load of a machine tool f_3 and the total load of a machine tool f_4 can be expressed as follows:

$$f_1 = \max C_i \tag{1}$$

$$f_2 = \sum_{i=1}^{N} \max\{C_i - d_i, 0\} \tag{2}$$

$$f_3 = \max \sum_{i=1}^{N} \sum_{j=1}^{n_i} t_{ijk} X_{ijk} \tag{3}$$

$$f_4 = \sum_{k=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{n_i} t_{ijk} X_{ijk} \tag{4}$$

Among them, i =1,2,…, N ; k =1,2,…, M ; C_i is the completion time of the work-piece J_i ; d_i is the delivery time of job J_i ; t_{ijk} is the processing time of the process j for work-piece J_i on No. k Machine; X_{ijk} is the decision variable, if the process j of the work-piece J_i can be processed by No. k Machine tools, then X_{ijk} = 1, otherwise X_{ijk} =0.

Now, we use hybrid genetic algorithm based on improved DNA computation (DNA-GA) to solve multi-objective scheduling optimization problems.

III. ALGORITHM DESIGN AND TEST

From the calculation angle of view, single strands of DNA chain become a string, can use four characters alphabet ∑={A, T, G, C} to represent information encoding. Give every DNA element a binary code, set A=00, T=01, G=10, C=11, can easily realize the interchange between DNA base and binary code. At the same time, maintain the complementary corresponding relations between DNA bases. That is, the calculation model of DNA coding can also be built based on two binary numbers---0, 1. The different coding characters are equivalence one-to-one mapping [21, 24].

Further, sometimes DNA double-strand structure isn't suitable for direct combining with the chromosomes of genetic algorithm. However, the unique single-strand RNA structure, and the vertical inheritance of genetic information Adenine, Guanine, Cytosine and Uralic, combine RNA computation and genetic algorithm is possible, Which make a great development for RNA computation. Such as Cukars has solved a chess problem by using the molecular computation method based on RNA; Li has summarized all the possible about RNA sequences [20, 25].

A. Pareto Sorting

In most cases, similar to the optimal solution of the single-objective doesn't exist in multi-objective optimization problems, only exists Pareto optimal solution set. The Pareto optimal solution of multi-objective problems is just an acceptable non-inferior or satisfactory solution. Usually, most of the multi-objective problems have many Pareto optimal solutions, in other words, the Pareto optimal solution of a multi-objective problem is a set. For practical application problems, on the basis of the understanding level of problems and decision-makers of personal preferences, select one or more solutions from Pareto optimal solutions of multi-objective, as the optimal solution of multi-objective. Therefore, find Pareto optimal solutions as much as possible. It is the most important step and the key for a multi-objective optimization problem.

Non-dominated sorting algorithm based on Pareto has applied successfully in multi-objective optimization algorithm. This algorithm can set all non-dominating solutions as 1 in the colonies; the other solutions are 2 or 3 in turn. In this paper, to guarantee the diversity of solutions, and avoid detection points concentrating on some optimal points. Use Pareto optimal solutions; put the individual density information into Pareto sorting

values. Calculations of the individual density values are shown as in Figure.1.

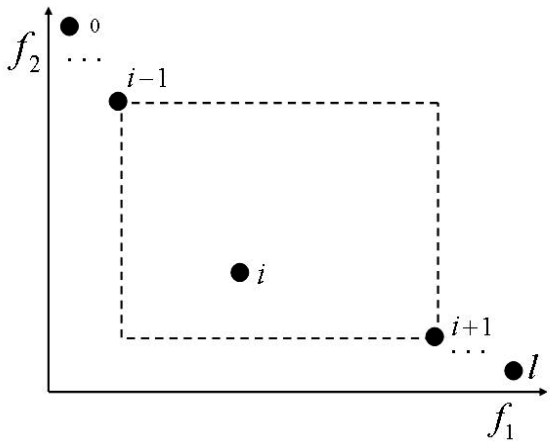


Figure 1. Calculation chart of individual density value

In Fig.1, suppose that when in the t generation of evolutionary algebra, individual i is in the furthest front of Pareto, Pareto sorting value is 1, then in the current Pareto front combination, the combination volume is $l+1$. At both ends of the individual are 0 , 1, as there were no other individuals on one of its side, so individual density is 0. The middle density value can be calculated as follows:

$$d_i^{'} = \frac{1}{\sum_{j=1}^{k}(f_j^{i+1} - f_j^{i-1})} \tag{5}$$

According to the formula (5), in the same Pareto-front, the more intensive individual distribution, the higher-density value it has. Otherwise, the sparser individual distribution, the smaller density value it has. First make a comparison, then normalize it, make $d_i = d_i^{'} / d_{\max}$, and $d_{\max}$ is the largest value of the current $d_i^{'}$. Therefore, the density information can be included in Pareto sorting value, therefore, the integrative fitness value of the current individual i is shown as follows:

$$F_{fit}(i) = i_r + \lambda d_i \tag{6}$$

In formula (6), parameter λ is to avoid $F_{fit}(i)$ and i_{r+1} has the same value.

B. Initialization and Coding

In a standard genetic algorithm, the initialization of chromosome is usually conducted through a random generation, which produced unlawful chromosome easily. In this paper, first, the RNA population initialization produced a number M of $N \times M$ matrices at random. Then, used the chromosome of natural numbers coding of [1, N] as RNA colonies formed by initial populations, the numbers of RNA populations are just M , which is different from the standard genetic algorithm. To prevent producing unlawful chromosome, used random numbers to produce the natural numbers between [1, N] in turn, recorded the times of each production, set each random number only appeared M times in chromosome, otherwise, reinitialized the population. In this way, unlawful chromosome can be prevented.

The solutions space of RNA sequences are E={A, U, G, C}^{L}. That is, RNA sequences used A, U, G, C four bases to encode a length of L RNA sequences, which composed by four kinds of bases: Uralic, Cytosine, Adenine, Guanine. Used 0(00), 1(01), 2(10), 3(11) to encode A, U, G, C with four-digit-system, then, the spaces of the L length RNA molecule sequences are E = {0, 1, 2, 3}^{L}.

Adopt coding ways based on an improved process. First, arrange scheduling code corresponding to working procedure, thus, each gene in the sequences of RNA population stand for one process. Can appoint all processes of a work-piece with the same symbol, compile these symbols according to which appeared in the order of the sequences, scan the sequences from left to right. The serial-number of a work-piece appeared i times stand for the number S_i process of the work-piece. So, for number N processes and number M machines, one RNA sequence includes a number $N \times M$ of genes, each work-piece appears M times in the sequence. To distinguish from the standard genetic algorithm, each gene doesn't stand for the specific process S_i of a work-piece i , but indicates the upper and lower dependency process. This way can overcome the possible process incompleteness in a work-piece process; make the random arrangement of RNA sequences always can produce feasible scheduling.

C. Fitness Calculation and Selection Operation

According to fitness calculation of the formula (6), define all individuals meet $F_{fit} < 2$ are elite individuals and preserve them. At the same time, suppose the largest value of reservation individuals is $N^{'}$. If the reservation number is larger than $N^{'}$, maintenance of the elite is necessary, so their distribution can represent evenly the whole Pareto optimization's assemblages.

The strategy for preserving elites is: first, if the elite individuals participate and the new solutions dominate part of the elite population solutions, wipe off the dominated solutions and add some new solutions. After reservation elite, the individuals in preserved colonies are all non-dominated individuals, their Pareto sorting values both are 1. When there are excessive preserved individuals, under the restriction of density grid, should delete high grid density to ensure only one individual for one grid. The objective function as dimensionality of the cell width is shown as follows:

$$g_{wi} = \frac{\max f_i(x) - \min f_i(x)}{K_i} \tag{7}$$

In formula (7), parameter g_{wi} is the width of the i-dimension cell. Parameter K_i is the quantity of the i-dimension cell, for the different evolutionary algebra. The largest and smallest values of objective function are quite different. The width of cell depends on the evolutionary algebra, but the quantity of cell will never change. Get the grid information, select the individuals in the same grid, and delete the dense individuals.

Through operation of elite reservation, the reserved colonies obviously are the best parents' colonies, can directly regard them as genetic operation of the parents. If the reserved colonies quantity is $N' < N$, according to the fitness value of the remaining colonies, use the championship method to choose the excellent individuals as the parents.

D. Crossover Operation

First, define the best $N/2$ individuals as neutral individuals, the remaining as harmful ones. Perform crossover operators both in neutral individuals, its probability of replacement operation is 1, and transposition operation is 0.5. Then, define R_2 as the random son-sequence within the range of the current sequence $[1, L]$, R_2' is the random son-sequence within the other neutral sequence, the sequence length of R_2' is the same as R_2. Perform the translocation operation, gain randomly R_2 from the former half part of the current sequence, generate randomly the new location of R_2 from the range of $[R_{2h} + L/2, L$, R_{2h} is a son-sequence in the latter part of the sequence which corresponds to the R_2 son-sequence. If it doesn't carry out the transposition operation, implement the replacement operation. Through transposition operation, generate randomly son-sequence R_2 from the former half part of the current sequence. Generate randomly son-sequence R_4 from the latter half part of the current sequence, which is the same length with R_2. Through crossover operation, generate N son sequences from $N/2$ neutral parents' sequences.

E. Mutation Operation

To maintain the population diversity and generate new genetic information, carry out mutation operation between the harmful parents' individuals and the sons' individuals which are produced by crossover operation, so there are $3N/2$ parents' individuals through implementing a mutation operator. For RNA-GA, RNA sequence should have many different hot spots and cold spots in the different stages of evolution. Then, mutation probability should be a dynamic process. Suppose the $[1, L/2]$ RNA sequence is the low position, $[L/2, L]$ is the high position, accordingly, define two kinds of mutation probability. High mutation probability is p_h and low mutation probability is p_l, which are shown as follows respectively:

$$p_h = a_1 + \frac{b_1}{1+\exp[\beta(g-g_0)]} \tag{8}$$

$$p_l = a_1 + \frac{b_1}{1+\exp[-\beta(g-g_0)]} \tag{9}$$

In formulas (8) and (9), a_1 is the final mutation probability of p_h, and the initial mutation probability of p_l. b_1 is the range of mutation probability, g is the current evolutionary algebra. g_0 is the turning point of hot spots and cold spots, and β is the change rate. Perform the above-mentioned mutation probability calculations, create L random numbers between [0, 1], compare it with the mutation probability, if the mutation probability is greater than the corresponding random number, perform the mutation operation.

F. Algorithm Steps

Step 1: set the population scale is N, the biggest evolutionary algebra is G, the grid numbers of the No. i target function is K_i;

Step 2: randomly generate initial quaternary sequences, calculate the individuals' fitness value;

Step 3: perform the operation of elite reservation and individual maintenance;

Step 4: retain the individuals as the paternal sequences of genetic operation, if the number of reservation individuals is less than N, for the shortage individuals. Apply the championship method to select the remaining individuals;

Step 5: perform the RNA crossover and mutation operator based on improved DNA computation;

Step 6: for the individuals which is generated in step 5, select the best $N/2$ sequences and the worst $N/2$ sequences as the initial populations of next generation, Calculate their fitness values. If the results meet the end condition, stop the operation; otherwise, jump to Step 3, repeat it until which meets the end condition.

G. Simulation Test

To test the effectiveness of the proposed algorithm, select a typical test function is shown as follows:

$$f(x) = x_1 \times \sin(\sqrt{|x_2 + 1 - x_1|}) \times \cos(\sqrt{|x_2 + 1 + x_1|}) + (x_2 + 1)\cos(\sqrt{|x_2 + 1 - x_1|}) \times \sin(\sqrt{|x_2 + 1 + x_1|})$$
$$x_1, x_2 \in [-512, 512] \tag{10}$$

In formulas (8), the optimal point value is -512, -512. The optimal solution is -511.701.

Suppose the designed algorithm and the SGA (standard genetic algorithm, SGA) adopt the same parameters. In several times of a simulation optimization process, find the convergence rate and optimizing performance of RNA-GA algorithm are more sensitive to a_1 and b_1, but not to β. Set up a larger a_1 and b_1 can speed up the convergence rate. However, if a_1 and b_1 parameters are too large, which will turn into a random searching algorithm, reduce the algorithm convergence rate. Similarly, if a_1 and b_1 parameters are too small, it is easy to fall into a local minimum point, also affect the algorithm performance. According to many times of simulation optimization results, can set the RNA-GA parameters a_1 between [0.01, 0.05], and b_1 between [0.10, 0.30] respectively, the experimental results are shown in Figure.2, Figure.3 and Figure.4.

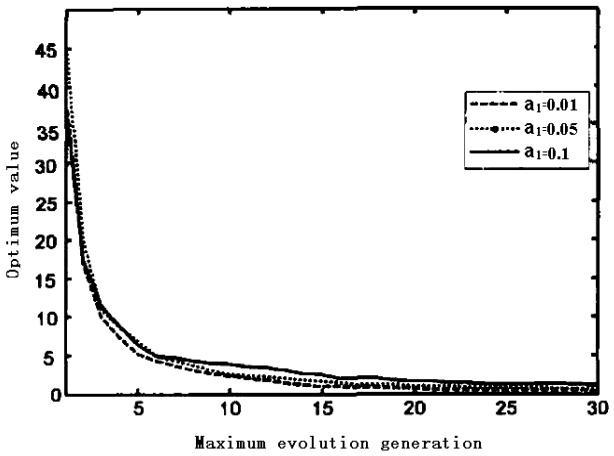


Figure 2. Convergence curve of a_1 parameter for different values

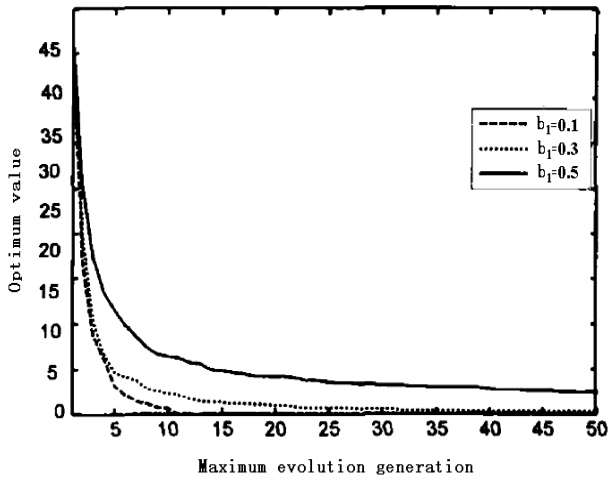


Figure 3. Convergence curve of b_1 parameter for different values

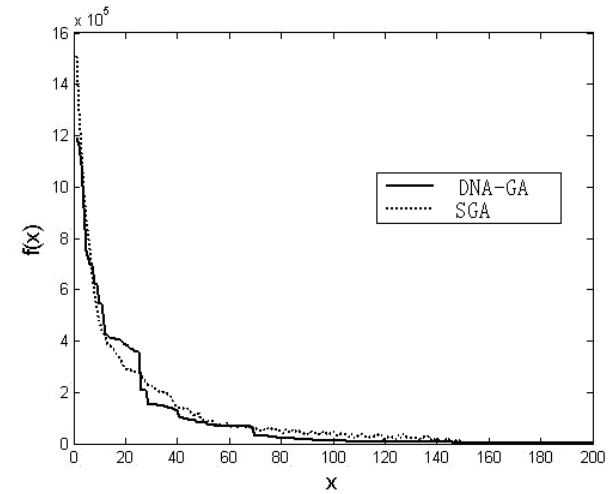


Figure 4. Convergence curve between DNA-GA and SGA

According to Figure.2, Figure.3 and Figure.4, in the beginning, DNA-GA and SGA both have similar initial population distribution. However, throughout evolution, DNA-GA has better diversity of the population than SGA, at the end of the proposed algorithm, concentrate most of the population in the optimal point, there have some population distribute beyond this point. This distribution structure will help to avoid the local optimal solutions, overcome the deceptive test function. However, SGA is a trend for some local solutions progressively, at the end of evolution, almost concentrates in a certain point. So, DNA-GA can better maintain the diversity of the population than SGA, has better search performance in the similar initial conditions.

For the objective function, the global optimal solution by DNA-GA obtained. Its location is at -488.65, 512 and the value is -511.72, which is better than the results given in the literature. Although DNA-GA search the global optimum probability is still lower, but in terms of relative SGA, this result has been greatly improved. In addition, as the objective function doesn't have model deceptiveness, both can find the global optimum. However, the optimal solution obtained by DNA-GA is obviously better than SGA.

IV. EXPERIMENT AND ANALYSIS

A. Experiment

To verify the above designed algorithm, use the following simulation test conditions: there are six machine tools to process four work-pieces, each work-piece has three processes, and each process can select more than one machine tool for processing. The processing time of each process is different on different machine tools. The processing time of a work-piece in each process on different machine tools are shown in Table. 1.

TABLE I.
Work-piece processing schedule

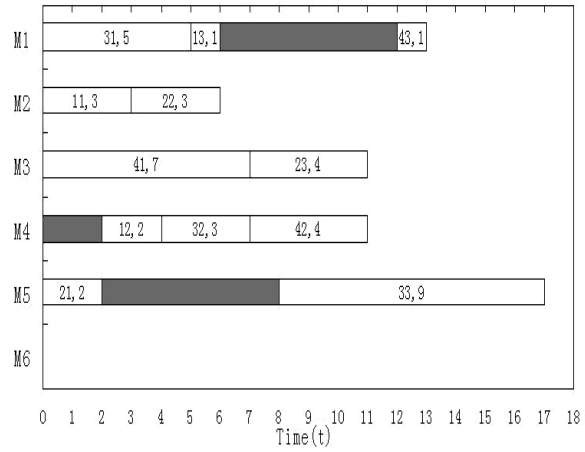
Work-piece	Process	M1	M2	M3	M4	M5	M6	Delivery time
	0	2	3	4				
J_1	1		3		2	4		14
	2	1	4	5				
	0	3		5		2		
J_2	1	4	3			6		17
	2			4		7	11	
	0	5	6					
J_3	1		4		3	5		13
	2			13		9	12	
	0	9		7	9			
J_4	1		6		4		5	12
	2	1		3			3	

Perform the simulation optimization operation. Set the largest evolutionary algebra G is 100, population number N is 50, sequence length L is 40, a_1 is 0.02, b_1

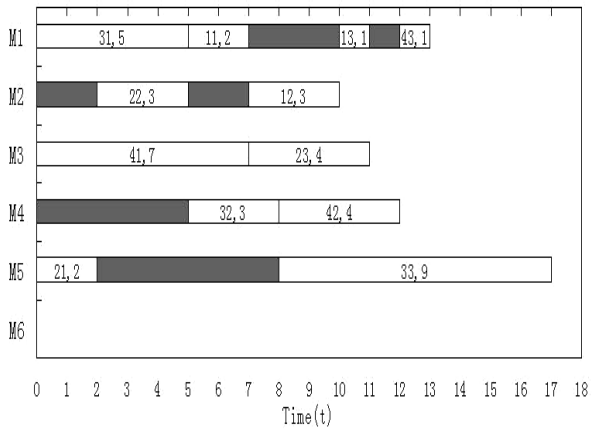
is 0.20, g_0 is G/2, β is 20/G, K_i is 50, λ is 0.9. The convergence results are shown in Fig.4, Obtain the Pareto optimal solution sets by two methods, which are shown in Table.2.

TABLE II.
Pareto optimal solution sets

Group	SGA				DNA-GA			
No.1	17	5	11	45	17	5	11	43
No.2	17	5	11	45	17	5	11	43
No.3	17	6	11	47	17	6	11	44
No.4	17	6	11	47	17	6	11	44
No.5	18	10	10	55	18	10	10	49
No.6	18	10	10	55	18	10	10	49



(a)



(b)

Figure 5. Gantt of two group optimal solutions obtained

Through analyzing the data in Table.2 and Table.3, under the same conditions, obtain the optimal solutions by RNA algorithm are obviously better than the standard genetic algorithm. Choose the two former optimal solution sets to make analysis, obtain the Gantt of scheduling results are shown in Figure. 5, the best route is shown in Table. 3.

Assign equally the processing task to each parallel machine, ensure that at least one machine is processing during the entire processing period, which indicates the scheduling results are reasonable. Although the processing orders of each work-piece are different, the optimal solutions for two groups can exactly get the same target values. So the designed algorithm has its realistic meaning for flexible scheduling optimization, which can expand the application of improved DNA computation.

TABLE III.
The best route

Work-piece	Operation	Optional machine					
		1	2	3	4	5	6
J_1	O11		3				
	O12			2			
	O13	1					
J_2	O21						2
	O22		3				
	O23						4
J_3	O31	5					
	O32				3		
	O33					9	
J_4	O41			7			
	O42				4		
	O43					1	

B. Comparison and analysis

Ponnambalam etc.[25] studied the job scheduling problem, used the production cycle, work-piece delays, machine idle time etc. as the targets, the classic scheduling problem "FT06" includes six work-pieces and six machine tools, Ponnambalam etc. got the best schedule is: production cycle is 76 minutes, the work-piece delay time is 31 minutes, machine idle time is 259 minutes. To confirm the designed algorithm, shield the largest load target of machine tool. The test result is: production cycle is 54 minutes, the work-piece delay time is 27 minutes, machine idle time is 131 minutes. The result is better than the former, the optimal scheduling Gantt chart is shown in Figure 6.

Analyze the results of simulation and comparison, can know biological RNA molecule computation replaces the realization processing of RNA-GA. Even delete the better solutions of RNA molecule calculation unconsciously, also can impose restructuring and mutating operation on RNA molecules, and get a better optimal solution in the next step. In other words, through the limited repeating operations, the optimal or second-best solutions will be greatly increased.

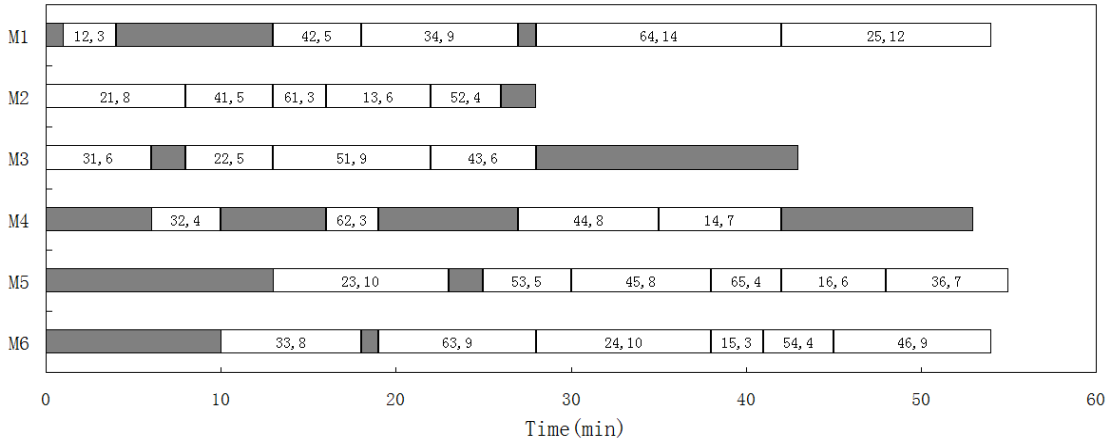


Figure 6. The optimal scheduling Gantt chart of FT06

V. CONCLUSIONS

To solve the problems of multi-objective flexible scheduling optimization, put forward a new Pareto non-dominated sorting genetic algorithm based on improved DNA computation. Used non-dominated sorting method and elite-reservation strategy, adopted quaternary encoding based on improved DNA computation and RNA genetic operators. Can ensure the diversity of solutions and achieve the multi-objective flexible scheduling optimization. Simulation results prove the feasibility of the designed algorithm, indicated it can provide a better searching and seeking abilities.

ACKNOWLEDGMENT

This work was supported by Project 50875086 of the National Science Foundation of China.

REFERENCES

[1] Beatrice Ombuki, Brian J. Ross and Franklin Hanshar, "Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows", *Applied Intelligence*. vol. 24, No. 1, pp. 17-30, January 2006.

[2] Fulya Altiparmak, Mitsuo Gen, Lin Lin and Turan Paksoy, "A genetic algorithm approach for multi-objective optimization of supply chain networks", *Computers & Industrial Engineering*. vol. 51, No. 1, pp. 196-215, September 2006.

[3] Yang Kaibing, "Multi-Objective Hybrid Genetic Algorithm for Flow Shop Scheduling", *Computer and Information Technology*. vol. 16, No. 2, pp. 28-30, February 2008.

[4] F. Pezzella, G. Morganti and G. Ciaschetti, "A genetic algorithm for the Flexible Job-shop Scheduling Problem", *Computers & Operations Research*. vol. 35, No. 10, pp. 3202-3212, October 2008.

[5] Yi-Tung Kao and Erwie Zahara, "A hybrid genetic algorithm and particle swarm optimization for multimodal functions", *Applied Soft Computing*. vol. 8, No. 2, pp. 849-857, March 2008.

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

[7] J.J.M. Mendes, J.F. Gonçalves and M.G.C. Resende, "A random key based genetic algorithm for the resource constrained project scheduling problem", *Computers & Operations Research*. vol. 36, No. 1, pp. 92-109, January 2009.

[8] ZHU Yu, ZHANG Hong, KONG Ling-dong, "Multi-dimension and Multi-level Association Rule Mining Based on Immune Genetic Algorithm", *Computer Engineering*. vol. 35, No. 23, pp. 181-183, December 2009.

[9] Zhu Sheng, Wang Yong-Ming, Hu Xiang-Qun, "Application of immune genetic algorithm to back analysis for parameters in model of rockfill dam coarse grain material", *Yantu Lixue (Rock and Soil Mechanics)*. vol. 31, No. 3, pp. 961-966, March 2010.

[10] Liu Xikui, "The Study on Coding and Some Optimization Models of DNA Computing and Genetic Algorithms", *Doctoral dissertation of Huazhong University of Science&Technology*, pp. 27-36, December 2003.

[11] Huang Buyi, "The Research on Several Theoretic Problems of DNA Computer", *Doctoral dissertation of Huazhong University of Science & Technology*. pp. 43-51, June 2005.

[12] Tao Jili, "Application and research of genetic algorithms based-on DNA computation", *Doctoral dissertation of Zhejiang University*, pp. 16-38, June 2007.

[13] NIE Shuzhi, "Research on Collaborative Manufacturing Resources Optimization Deployment based on DNA genetic algorithm", *Doctoral dissertation of South China University of Technology*, pp. 75-92, June 2010.

[14] W Liu ,S Wang ,J Xu, "A new DNA computing model for the NAND gate based on induced hairpin formation", *BioSystems*. vol. 177, No. 3, pp. 87-92, March 2004.

[15] Li S. C., J. Xu, "Digtal Coding for RNA based on DNA Computation", *Computer Engineering and Application*. vol. 39, No. 5, pp. 46-47, February 2003.

[16] Liu Yi, Ye Chunming, ShenYunhong, "The DNA Genetic Algorithm Applied for Flow Shop Scheduling Problem", *Computer Engineering and Applications*. vol. 41, No. 17, pp. 85-87, June 2005.

[17] Guohui Zhang, Xinyu Shao, Peigen Li, Liang Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem", *Computers & Industrial Engineering*. vol. 56, No. 4, pp. 1309-1318, May 2009.

[18] Ghasem Moslehi, Mehdi Mahnam, "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search", *International Journal of Production Economics*. vol. 129, No. 1, pp. 14-22, 2010.

[19] Jun-qing Li, Quan-ke Pan, Yun-chia Liang, "An effective hybrid tabu search algorithm for multi-objective flexible

job shop scheduling problems", *Computers & Industrial Engineering*. vol. 59, No. 4, pp. 647-662, November 2010.

[20] N. Karimi, M. Zandieh, H.R. Karamooz, "Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach", *Expert Systems with Applications*. vol. 37, No. 6, pp. 4024-4032, June 2010.

[21] Catarina Dudas, Marcus Frantzén, Amos H.C. Ng, "A synergy of multi-objective optimization and data mining for the analysis of a flexible flow shop", *Robotics and Computer Integrated Manufacturing*. vol. 27, No. 4, pp. 687-695, 2011.

[22] Voratas Kachitvichyanukul and Siriwan Sitthitham, "A two-stage genetic algorithm for multi-objective job shop scheduling problems", *Journal of Intelligent Manufacturing*. vol. 22, No. 3, pp. 355-365, 2011.

[23] Liu Xiao-xia, Xie Li-yang, Tao ze, Hao Chang-zhong, "Flexible Job Shop Scheduling Optimization of Multi-objective", *Journal of Northeastern University(Natural Science Edition)*. vol. 29, No. 3, pp. 362-365, March 2008.

[24] Li-Ning Xing, Ying-Wu Chen, Ke-Wei Yang, "Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling", *Applied Soft Computing*. vol. 9, No. 1, pp. 362-276, January 2009.

[25] Ponnambalam S G, Ramkumar V, Jawahar N, "A multiobjective genetic algorithm for job shop scheduling", *Production Planning & Control*. vol. 12, No. 8, pp. 764-774, 2001.

NIE Shuzhi Hunan Province, China. Birth date: November, 1977. is Mechanical Manufacturing and Automation Ph.D., graduated from Mechanical and Automotive Engineering College, South China University of Technology. And research interests on advanced manufacturing technology, intelligent computing and simulation optimization.

ZHONG Yanhua Guangdong Province, China. Birth date: October, 1974, is Software Engineering M.S., graduated from Computer College, Guangdong University of Technology, China. And research interests on intelligent algorithms and quantum compuing. She is an associate professor.