Improved Niche GA for FJSP

Jingjing Liang[1], Qin Wang[1,2], Wenxing Xu[1*], Zisen Gao[1], Zheng Yan[1], Fengling Yu[1]

[1]College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China
[2] College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China
xuwenxing@bipt.edu.cn

Abstract: When solving the flexible job shop scheduling problem (FJSP), the optimization performance of the traditional genetic algorithm is often degraded in the late stage of population evolution because of the reduction of the diversity of the population. In this paper, an improved niche genetic algorithm (INGA) was proposed to minimize makespan for solving FJSP. This new algorithm uses a pre-selection mechanism based on the niching technology, and guarantees the optimization performance and population diversity of the algorithm. The population diversity of INGA and the traditional genetic algorithm for solving FJSP is compared using the population independence ratio evaluation indicator. Benchmarking instance studies verify that INGA has great advantages in solving FJSP. Furthermore, main contribution of the paper is extended with the comparison of two sets of practical engineering application cases proving the effectiveness of the adopted INGA for the resolution of combinatorial optimization problems.

Keywords: flexible job shop scheduling; niching technology; pre-selection mechanism; population diversity; genetic algorithm

1 Introduction

FJSP is well known to be NP-hard[1-2]. Due to the computational complexity of FJSP, various meta-heuristics have been extensively applied to solve the challenging FJSP. Such proposals include Artificial Bee Colony Algorithms [3], Particle Swarm Approaches [4], Genetic Algorithm[5-6], Artificial Fish Swarm Algorithm[7], Bat-inspired algorithm[8], Grey Wolf Algorithm[9], Ant Colony Algorithm[10]. In the process of solving FJSP, the individual experience in the population falls into the local optimum, which will decrease the population diversity, reduce the coverage of the solution, then increases and the difficulty of finding the global optimum. Therefore, how to ensure the population diversity in the process of population evolution is the key problem to avoid the population optimization algorithm falling into local optimum [11].

In order to solve the problem of population diversity reduction in the population optimization algorithm, the researchers proposed to solve the FJSP by combining different evolutionary algorithms to maximize the advantages of the algorithm. For example, Hybrid of Genetic Algorithm/Simulated Annealing Algorithm [12], Hybrid of Immune Algorithm/Genetic Algorithm [13], Hybrid of Bee Algorithm/Simulated Annealing approach [14], Hybrid of Bee Particle Swarm/Artificial Immunity approach [15]. However, this method increases the computational complexity of solving FJSP, the algorithm is not easy to implement, and the accuracy of the solution is also poor. Then the researchers applied the biological niche ideas to the niche technology in the optimization algorithm, which better solved the problem of population diversity reduction in the population optimization algorithm [16].Therefore, in order to solve FJSP, the focus of this paper is to integrate niching technology into genetic algorithm, and use niching technology based on pre-selection to pre-select the updated individuals to maintain the diversity of the population. Avoiding the late stage of evolution, individuals with high fitness occupy the entire population, making the genetic algorithm enter the optimal stagnation state due to the intersection and variation between close relatives.

2 INGA

2.1 A pre-selection mechanism based on improved niching technology

The replacement rule stipulated by the pre-selection strategy[17] proposed by the researcher Cavicchio in 1970 only considers the chromosome structure of the population, so that it remains as constant as possible during the evolution of the population, and thus maintains population diversity. However, such strict replacement rules will lead to the weakening of GA optimization performance. Considering the diversity of population in evolution and the optimization performance of GA, this paper redefines the replacement rule in the reference [17], that is, each child of the current population can not only be replaced with its immediate parent, but also with other parents of the last generation. The pre-selection mechanism based on the niching technology proposed in this paper regards all chromosome individuals(P) in the paternal generation as P types of niches. Hamming distances are calculated one by one between each chromosome produced by the children and all the chromosomes of the parents, then the child and the parent chromosome closest to the Hamming distance are selected by the fitness value as the evaluation indicator. when the indicator values are consistent, the algorithm selects the children who have undergone the genetic update operation.

978-1-7281-3863-3/19/$31.00 ©2019 IEEE

In this paper, vector $X_i = \left(C_1^i, C_2^i, \ldots C_{To}^i, \ldots, C_{2\times To}^i\right)$ is used to represent the ith chromosome of the contemporary population, vector $X_j = \left(C_1^j, C_2^j, \ldots C_{To}^j, \ldots, C_{2\times To}^j\right)$ is used to represent the jth chromosome of the contemporary population. The sum Hij of the absolute values of the differences between the genes at the same position of the Xi and Xj chromosomes is defined as the Hamming distance, the formula for Hij is as follows:

$$H_{ij} = \sum_{k=1}^{2\times To} \left|C_k^i - C_k^j\right| \tag{1}$$

The specific implementation steps of the pre-selection mechanism based on the niching technology proposed in this paper are as Figure 1:

```
Input: ParentX, TempX
Output: ChildrenX
1-For each element i of TempX, with i=[1,…P] do
2-  Chromosome_1= TempXi
3-  For each element r of ParentX, with r=[1,…P] do
4-    Calculate the Hamming distance between Chromosome_1 and ParentXr
5-    Record the Hamming distance value and its chromosome
5-  EndFor
6-  Select Chromosome_2 equal=chromosome with min Hamming distance value,
7-  Record location number of the Chromosome_2(minH_Index)
8-  Decoding Chromosome_1 and Chromosome_2
9-  Calculate their makespan(makespan_1 and makespan_2)
10- If makespan_1   makespan_2,
11-   Update ChildrenXminH_Index into Chromosome_1
12- Else
13-   ChildrenX unchanged
14- EndIf
15-EndFor
```

Figure 1 Improved niche-based pre-selection mechanism

2.2 The overall steps of the INGA

Based on the above pre-selection mechanism, the INGA is proposed in this paper. In the initialization mechanism, the machine selection is generated by the GRS initialization mechanism [18], and the operations sequencing is generated by a random approach; Selection operation adopts tournament selection strategy; For crossover operation, the machine selection adopts the uniform crossover [19] operation, and the operations sequencing adopts the POX crossover [19]. For mutation operation, the machine selection adopts the single point mutation [19], and the operations sequencing adopts the neighborhood search mutation [19]; Decoding process by inserting decode [19] approach. The framework of the algorithm is shown in Figure 2.

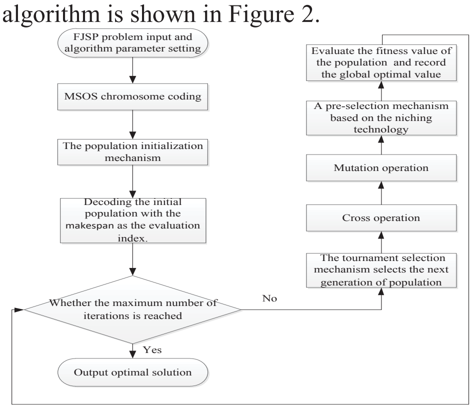

Figure 2 INGA solves FJSP flow chart

3 Analysis of population diversity

3.1 Indicator of population diversity

Population diversity in population evolution can be directly reflected in population size and the number of individuals. In this paper, Nindependent is used to represent the total number of independent individuals in the population; Ntotal is used to represent the total number of all individuals in the population. And the ratio of independent Nindependent to Ntotal is defined as the population independent ratio, expressed as w=Nindependent/Ntotal,and its value range is 0≤w≤1.If w=1, it means that all individuals in the population are independent individuals; if w=0, it means that all individuals in the population are the same.

Each chromosome is regarded as a vector in this paper, and the number of Nindependent can be obtained by calculating the corresponding vector norm. In addition, a chromosome similarity factor is proposed, and the vector norm is adopted to calculate the similarity between different chromosome individuals.For any vector $x = \left(x1, x2, \ldots,\right)$, the one-dimensional norm of the vector is $\|x\|_1 = \sum |x_i|$. Let the vector $I_k = \left(C_1^k, C_2^k, \ldots C_{To}^k, \ldots, C_{2\times To}^k\right)$ denote the kth chromosome individual of the contemporary population, and let the vector $I_l = \left(C_1^l, C_2^l, \ldots C_{To}^l, \ldots, C_{2\times To}^l\right)$ denote the lth chromosome individual of the contemporary population. To is the number of total operations in the scheduling system, while 2xTo represents the total length of the chromosomes in each individual population. δkl can express the chromosome similarity between the individuals represented by the kth chromosome and the lth chromosome.in the whole population. When δkl =0, it means that the two groups of individuals are identical; When δkl≠0, it means that the two groups of individuals are not identical, while the larger the value of δkl, the greater of difference between the two chromosomes. the formula is as follows:

$$\delta_{kl} = \left\|I_k - I_l\right\|_1 = \sum_{j=1}^{2\times To} \left|C_i^k - C_i^l\right| \tag{2}$$

In this paper, ηkl is used to represent the difference flags of the two individuals corresponding to the kth and lth chromosomes. When the two individuals are identical, ηkl =1; when the two individuals are not identical, ηkl =0, the formula is as follows:

$$\eta_{kl} = \begin{cases} 1, when\ \delta_{kl} = 0; \\ 0, when\ \delta_{kl} \neq 0. \end{cases} \tag{3}$$

The formula for calculating the number of independent individuals of the total population is as follows:

$$N_{independent} = N_{total} - \sum_{k=1}^{P-1} \sum_{l=k+1}^{P} \eta_{kl} \tag{4}$$

3.2 Population diversity analysis experiment

In this section, the comparative experiment adopts the improved genetic algorithm in [18] and the INGA with the niching technology proposed in this paper to solve the Brandimartre [1] benchmarking instance MK01. The main parameters of the algorithm are set as follows: Population size P=100, maximum iteration number Gm=200, crossover probability Pc=0.8, mutation probability Pm=0.01, continuous operation times COT=1, the condition for exiting the loop is to reach the maximum number of iterations Gm, the optimization goal is to minimize the makespan required for MK01.

The improved GA and INGA solve the FJSP standard test instance MK01, and the variation curve of the population diversity during the whole iteration process is shown in Figure 3.

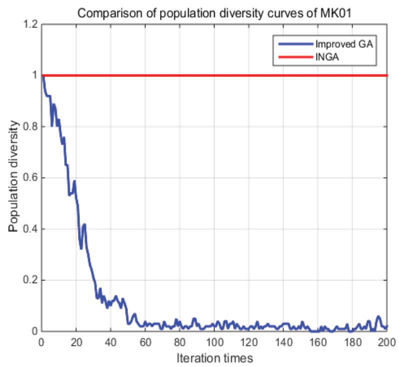

Figure 3 Population diversity curve of Improved GA and INGA when solving MK01

The former is marked with blue image, and the latter is marked with red image. Analysis of Figure 3, improved GA obtained w from 1 to 0 slowly, INGA obtained w is always 1. This shows that the quality of initial population generated by improved GA is high, so the improved GA has a strong guiding significance for solving FJSP problem. However, as the iteration proceeds, the diversity is quickly lost, so it does not solve the shortcoming of traditional genetic algorithm that easily falls into local optimal solution. In contrast, the introduction of niche-based pre-selection mechanisms can effectively avoid this situation, so that the improved algorithm maintains good population diversity throughout the solution process.

4 Benchmarking instances comparison experiments and analysis

4.1 Comparison experiment 1

The optimal value of the maximum scheduled completion time(Cmax), the average value of the optimal value(AVCmax), the worst value of the optimal value(MaxCmax), and the variance value of the optimal value(VarCmax) are experimental evaluation indicators. In this section, different algorithms are used to solve FJSP benchmark instances designed by Brandimartre [1] and Kacem [20] with the same parameter. The purpose of this experiment is to test the performance of INGA in solving FJSP. The basic parameters of each algorithm are set as follows: P=100, Gm=200, Pm=0.8, Pc=0.01, COT=10. After a lot of experiments, when Pm=0.8 and Pc=0.01, the algorithm can achieve better results.

The comparison results of Cmax related indicators obtained by different algorithms in the FJSP standard test instances with the same parameter are shown in Table I. Analyzing Table I, the random experiments of solving eleven benchmarking instances ten times in succession. The best value is marked in bold.The results obtained by INGA to solve FJSP are compared with the first three groups of experiments, and 11/11 groups of instances obtained the optimal value of Cmax, the better rate reached 100%, 9/11 groups got the smallest AVCmax, 10/11 groups got the smallest MaxCmax, and 8/11 groups got the smallest VarCmax. When solving the FJSP benchmarking instances MK02, MK04, MK05, MK07, the indicators of the maximum scheduled completion time have been significantly improved. This shows that INGA has strong global optimization, high reliability and strong robustness in solving FJSP.

Inst.	Reference[19]				Reference[21]				Reference[18]				INGA			
	C_{max}	AVC_{max}	$MaxC_{ma}$	$VarC_{max}$	C_{max}	AVC_{max}	$MaxC_{ma}$	$VarC_{max}$	C_{max}	AVC_{max}	$MaxC_{ma}$	$VarC_{max}$	C_{max}	AVC_{max}	$MaxC_{ma}$	$VarC_{max}$
mk01	42	42	42	**0.0000**	42	42	42	**0.0000**	40	**40.5**	41	0.2778	**40**	40.6	**41**	0.2400
mk02	29	30.3	32	1.3444	29	30.4	32	0.7111	28	28.9	30	0.5440	**28**	**28.5**	**30**	**0.4500**
mk03	**204**	**204**	**204**	**0.0000**	**204**	**204**	**204**	**0.0000**	**204**	**204**	**204**	**0.0000**	**204**	**204**	**204**	**0.0000**
mk04	**63**	66.1	67	1.8778	65	66.5	67	**0.5000**	65	65.8	67	0.6222	**63**	**65.6**	**67**	1.8400
mk05	177	179.1	181	2.9889	177	178.7	181	2.2333	176	178.6	181	2.0444	**174**	**176.7**	**179**	**1.8100**
mk07	145	150.6	156	11.1570	145	150	154	7.1111	145	147.1	149	2.3222	**142**	**144**	**145**	**0.6000**
mk08	**523**	**523**	**523**	**0.0000**	**523**	**523**	**523**	**0.0000**	**523**	**523**	**523**	**0.0000**	**523**	**523**	**523**	**0.0000**
mk09	**311**	**311.2**	**312**	**0.2667**	**311**	311.4	313	0.4889	**311**	314.3	322	14.233	**311**	312.5	314	1.2500
8x8	**14**	14.4	15	0.2667	**14**	14.4	15	0.4889	**14**	**14**	**14**	**0.0000**	**14**	**14**	**14**	**0.0000**
10x10	**7**	7.9	8	0.1000	**7**	7.9	8	0.1000	**7**	**7**	**7**	**0.0000**	**7**	**7**	**7**	**0.0000**
15x10	12	12.5	14	0.5000	12	13.0	14	0.2220	12	12.7	13	0.2333	**11**	**11.8**	**12**	**0.1600**

Table I Comparison of results of indicators obtained by FJSP with different algorithms under the same parameter

4.2 Comparison experiment 2

The optimal value of the maximum scheduled completion time (Cmax) and the average value of the optimal value(AVCmax) are experimental evaluation indicators. In this section, different algorithms are used to solve FJSP benchmark instances designed by Brandimartre [1] and Kacem [20] with different parameters, and the results of INGA solving FJSP are compared with the data in reference [18-19,21]. The main parameters of the genetic algorithm in the reference [19] and the reference [21] are set as follows: P=500，Gm=100，Pm=0.8，Pc=0.05，COT=10, when the number of runs reaches the maximum number of iterations or the global optimal value stagnates for thirty times, the loop ends. The experimental results are shown in the first two columns of Table II. The smaller the product of P and Gm, that is, the less the number of evaluation calculations of the algorithm, the higher the efficiency of the algorithm. Therefore, in the second experiment, the main parameter settings of the improved GA in [18] are used in the comparison test: P=100，Gm=200，Pm=0.8，Pc=0.01，COT=10，the condition for exiting the loop is to reach the maximum number of iterations. The experimental results are shown in the last two columns of Table II.

Table II Comparison of results of indicators obtained by FJSP with different algorithms under the different parameter

Inst.	*n x m*	Reference [19]		Reference [21]		Reference [18]		INGA	
		C_{max}	AVC_{max}	C_{max}	AVC_{max}	C_{max}	AVC_{max}	C_{max}	AVC_{max}
mk01	10x6	**40**	41.7	**40**	40.8	**40**	**40.5**	**40**	40.6
mk02	10x6	**28**	29.4	**28**	28.9	**28**	28.9	**28**	**28.5**
mk03	15x8	204	204.0	204	204.0	204	204.0	**204**	**204.0**
mk04	15x8	64	66.5	**63**	66.0	65	65.8	**63**	**65.6**
mk05	15x4	177	179.4	**174**	176.8	176	178.6	**174**	**176.7**
mk07	20x5	145	148.8	145	148.9	145	147.1	**142**	**144.0**
mk08	20x10	**523**	**523**	**523**	**523**	**523**	**523.0**	**523**	**523.0**
mk09	20x10	313	317.1	**311**	314.9	**311**	314.3	**311**	**312.5**
8x8	8x8	**14**	15.1	**14**	14.8	**14**	**14.0**	**14**	**14.0**
10x10	10x10	**7**	7.1	**7**	7.1	**7**	**7.0**	**7**	**7.0**
15x10	10x10	12	12.0	**11**	11.9	12	12.7	**11**	**11.8**

The comparison results of Cmax related indicators obtained by different algorithms in the FJSP benchmarking instances with different parameters are shown in Table II. Although the product of P and Gm of the algorithm adopted in this paper is obviously smaller than that in reference [19] and [21], it can be seen from the analysis of Table 2 that INGA has obvious superiority in solving FJSP. In a random experiment for eleven benchmarking instances proposed by Brandimartre [1] and Kacem [20], running ten times in succession. The results obtained by INGA to solve FJSP are compared with the first three groups of experiments, 11/11 groups of instances obtained the optimal value of Cmax and 10/11 groups got the smallest AVCmax. When solving the FJSP benchmarking instances MK04, MK05, MK07, MK09, The optimal value of the maximum scheduled completion time (Cmax) and the average value of the optimal value (AVCmax) have been significantly improved. According to the comparison results of the two indicator values of the four sets of experiments, INGA has strong global optimization ability and strong reliability when solving FJSP.

5 Engineering application

5.1 Case 1

The flexible job shop of a space company producing multi-species aerospace components as test case 1 [22]. Apply the INGA algorithm in the paper to solve case 1. The main parameters of the algorithm are set as follows:P=100, Gm=200，Pm=0.8，Pc=0.1, COT=10, with the maximum completion time (Cmax) as the evaluation indicator, the optimal value and average value of Cmax after continuous operation for COT times are shown in Table III.

Table III Comparison results of case 1

Problem	*To*	*n x m*	Reference [22]		INGA	
			C_{max}	AVC_{max}	C_{max}	AVC_{max}
Case 1	36	6x10	31	--	**29**	**29.9**

5.2 Case 2

Take the example of the processing workshop of an aeroengine company in the reference [23] as the test case 2, The main parameters of the algorithm are set as follows: P=100, Gm=50，Pm=0.8，Pc=0.1, COT=30, with the maximum completion time (Cmax) as the evaluation indicator, the optimal value, average value and the worst value of Cmax after continuous operation for COT times are shown in Table IV.

Table IV Comparison results of case 2

Algorithm	C_{max}	AVC_{max}	$MAXC_{max}$
GA	60	63.3	65
ACO	58	60.4	61.0
GA-ACO	53	54.8	55.0

MACO	53	54.1	54.5
INGA	**51**	**51.9**	**53**

6 Conclusions

When the traditional genetic algorithm solves the NP-hard of FJSP, a global or local optimal individual often appears in the late stage of population evolution, and the individual will soon occupy the entire population solution space as a super-individual, making the algorithm fall into a local excellent stagnation state, which is often caused by the low population diversity of the algorithm. In this paper, the population independence ratio is proposed as an evaluation indicator of population diversity, and it is proved that the population diversity has an important influence on the precision of genetic algorithm by experiments; a pre-selection mechanism based on the niching technology was proposed to prevent population diversity from being destroyed during evolution; the INGA is adopted to solve the FJSP with the minimum completion time of the scheduling as the evaluation indicator, and the superiority of INGA in solving FJSP is illustrated by the test of the standard example; then the improved algorithm is applied to the production scheduling system of two practical flexible job shops, which verifies that the improved algorithm has strong practicability and feasibility in solving practical application problems.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61304217, No.21703013) and the Cultivation Plan of Young Talents in Municipal Colleges and Universities (No.CIT&TCD201704048).

References

[1] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search[J]. Ann Oper Res,1993,41:157-183.

[2] Jain A S, Meeran S. Deterministic job-shop scheduling: Past, present and future[J]. European journal of operational research, 1999, 113(2): 390-434.

[3] Gao K Z, Suganthan P N, Chua T J, et al. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion[J]. Expert Systems with Applications An International Journal, 2015, 42(21):7652-7663.

[4] Singh M R, Mahapatra S S. A quantum behaved particle swarm optimization for flexible job shop scheduling[J]. Computers & Industrial Engineering, 2016, 93: 36-44.

[5] Li X, Gao L. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem[J]. International Journal of Production Economics, 2016, 174: 93-110.

[6] Zhang M, Wu K. An improved genetic algorithm for the re-entrant and flexible job-shop scheduling problem[C]//2016 Chinese Control and Decision Conference (CCDC). IEEE, 2016: 3399-3404.

[7] Alobaidi A T S, Hussein S A. An improved Artificial Fish Swarm Algorithm to solve flexible job shop[C]//2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT). IEEE, 2017: 7-12.

[8] Yang X S. A new metaheuristic bat-inspired algorithm[M]//Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Berlin, Heidelberg, 2010: 65-74.

[9] Jiang T H. Flexible job shop scheduling problem with hybrid grey wolf optimization algorithm[J]. Control and Desicion, 2018, 33(3): 503-508.

[10] Song D, Zhang J. Batch scheduling problem of hybrid flow shop based on ant colony algorithm[J]. Computer Integrated Manufacturing Systems, 2013, 19(7): 1640-1647.

[11] Wang J Q, Guo Y ZH, Cui F D, et al. Open shop scheduling optimization based on diversity enhanced adaptive genetic algorithm [J]. Computer Integrated Manufacturing System, 2014, 20 (10): 2479-2493.

[12] Huang, X B; Yang, LX, et al. A hybrid genetic algorithm for multi-objective flexible job shop scheduling problem considering transportation time [J]. International Journal of Intelligent Computing and Cybernetics, 2019, 12 (2). 154-174.

[13] Yuan C A I, Jinhua C. FLEXIBLE JOB SHOP FUZZY SCHEDULING METHOD BASED ON IMMUNE GENETIC ALGORITHM[J]. Academic Journal of Manufacturing Engineering, 2018, 16(4).

[14] Phu-ang A, Thammano A. Memetic algorithm based on marriage in honey bees optimization for flexible job shop scheduling problem[J]. Memetic Computing, 2017, 9(4): 295-309.

[15] A chaotic simulated annealing and particle swarm improved artificial immune algorithm for flexible job shop scheduling problem.

[16] Li Y, Chen Y, Zhong J, et al. Niching particle swarm optimization with equilibrium factor for multi-modal optimization[J]. Information Sciences, 2019, 494: 233-246.

[17] Manner R, Mahfoud S W. Crowding and Preselection Revisited[J]. R.männer & B.manderick Parallel, 1992:27-36.

[18] Xu W X, Wang Q, Bian W B, et al. Improved GA and global random machine selection based on key operation to solve FJSP [J]. Journal of Chemical Engineering, 2017, 68 (3): 1073-1080.

[19] Zhang G H. Research on methods for flexible job shop scheduling problems[D]. Wuhan: Huazhong University of Science and Technology, 2009.

[20] Kacem I, Hammadi S, Borne P. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problem[J]. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews,2002,32(1):1-13.

[21] Zhao S K, Fang SH Q, Gu X J. Machine selection and FJSP solution based on limit scheduling completion time minimization[J]. Computer Integrated Manufacturing Systems, 2014, 20(4): 854-865.

[22] Shao B B, Cai H M, Jiang L H. A fast heuristic scheduling algorithm for flexible workshops [J]. Computer Applications and Software, 2009, 26 (3): 32-34.

[23] Xue H Q, Wei SH M, Zhang P, et al. Research on Flexible Job Shop Scheduling Based on multi group ant colony algorithm [J]. Computer Engineering and Application, 2013, 49 (24): 243-248.