ATLANTIS PRESS

Advances in Computer Science Research, volume 86

International Symposium on Communication Engineering & Computer Science (CECS 2018)

A Block - Based Improved NSGA - II Algorithm for Solving Multi - Objective Permutation Flow Shop Scheduling Problem

Xiaobing Pei, Chunhua Zhang [a]

School of Management, Tianjin University of Technology, Tianjin 300384, China.

[a] Zhang_chunhua1203@126.com

Abstract. Aiming at the problem of minimizing the maximum makespan and minimizing the delay time of multi-objective permutation flow shop scheduling problem (MOPFSP), a block-based artificial chromosome non-dominated sorting genetic algorithm II (NSGA-II) is proposed. The algorithm combines the stochastic mechanism and the opposition-based learning mechanism to generate the initial solution to balance the diversity and quality of the initial population. An elite population is generated through the operation of several generations of NSGA-II, and a location matrix and a dependency matrix are established for the elite population by using ant information density. Based on the two matrix mining blocks, the blocks and non-blocks are recombined to form artificial chromosomes.The algorithm will be tested by Reeves instance and Taillard instance, and compared with the results obtained by other algorithms to verify the effectiveness of the algorithm.

Keywords: permutation flow shop scheduling problem; multi-objective optimization; NSGA-II; artificial chromosome.

1. Introduction and Literature Review

The permutation flow-shop scheduling problem (PFSP) is a combinatorial problem that has drawn much attention. it is of practical value to study Multi-objective Permutation Flow-shop Scheduling Problem (MOPFSP). Many scholars at home and abroad have studied MOPFSP. Huang Xia et al (2017a) proposed an improved chaos weed optimization algorithm. This algorithm uses gray entropy weighting relevance entropy assignment method and fast non-dominated sorting method to solve the MOPFSP with the goal of minimizing the maximum completion time, the total flow time and the total delay time. Deng et al. (2017d) proposed a competitive memetic algorithm (CMA) to solve MOPFSP. CMA uses two populations with different objectives, designs some operators for each object for each population and designs a special interaction mechanism between two populations. In addition, a competition mechanism is proposed to adjust adaptively the selectivity of operators, and a knowledge-based local search operator is developed to improve the searching ability of CMA. Rifai et al. (2016a) proposed a new multi-objective adaptive large-neighborhood search algorithm based on Pareto frontier to solve MOPFSP for minimizing the completion time, total cost and average delay.

Chang et al. (2013b) proposed a block-based artificial chromosome genetic algorithm (p-ACGA) to solve the single goal of minimizing the maximum completion time in the permutation flowshop scheduling problem. The algorithm combines ant colony algorithm and genetic algorithm to produce elite population through crossover and mutation of simple genetic algorithm, and uses the pheromone concentration of ant colony algorithm to analyze the relationship between the workpieces. The statistical information is used to establish the pheromone-dependent matrix and to mine the blocks according to the matrix. The artificial chromosomes are formed by block and non-block recombination.In this paper, the algorithm is improved, and a fast non-dominated genetic algorithm based on block (p-ACNSGA-II) is proposed to solve the multi-objective permutation flowshop scheduling problem. When initializing, the random mechanism and opposition-based learning mechanism are combined to improve the quality of the initial solution. The NSGA-II for solving multi-objective problems is improved. The non-dominated solution set is constructed by fast sorting to improve the running speed. The distribution function is introduced into elite retention strategy to improve the uniformity of solution. Further considering the relationship between the workpiece and the position of the solution sequence, ie.,the distribution of pheromone concentration on the nodes, which speeds up the evolution of the algorithm. Finally, the introduction of chromosome recombination mechanism to improve the quality of solution.

2. Multi-Objective Permutation Flow Shop Scheduling Problem

2.1 Multi-Objective Optimization Problem Description

Taking the objective function minimization as an example, the mathematical model (adopted from Gao et al. 2012a) is described as following:

$$\left.\begin{array}{c}\min y = F(x) = \{f_1(x), f_2(x), \ldots, f_k(x)\} \\ s.t. g_i(x) \le 0, i = 1,2,\ldots,h \\ x \in X \in R^n, y \in Y \in R^m\end{array}\right\} \tag{1}$$

Where x is the decision vector, including n decision variables $(x_1, x_2, \ldots, x_n)$; y is the target vector, which consists of k objective functions $(y_1, y_2, \ldots, y_n)$; X is the decision space formed by x. Y is the target space formed by y.

2.2 Permutation Flow Shop Scheduling Problem

The PFSP production scheduling problem can be described as: n jobs are processed on m machines in the same order, with the same order of work on each machine. At the same time, there are some important conditions for this problem:

1). the operation is independent and can start at zero time;

2). a maximum of one job can be machined at the same time per machine;

3). each job can be processed on only one machine at a time.

Let $P(i, j)$ denotes the processing time of workpiece i on machine j, (π_1, π_2, π_3,…, π_n) denotes the workpiece sequence, and $C(\pi_i, j)$ means the makespan. Equations are as follows:

$$C(\pi_1,1) = p(\pi_1,1) \tag{2}$$

$$C(\pi_i,1) = C(\pi_{i-1},1) + p(\pi_i,1), \text{ for } i = 2,\ldots,n \tag{3}$$

$$C(\pi_1, j) = C(\pi_1, j-1) + p(\pi_1, j), \text{ for } j = 2,\ldots,m \tag{4}$$

$$C(\pi_i, j) = \max\{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + p(\pi_i, j), \text{ for } i = 2,\ldots,n, \text{for } j = 2,\ldots,m \tag{5}$$

The completion time of the last job in the sequence on the last machine:

$$C_{\max}(\pi) = p(\pi_n, m) \tag{6}$$

Total flow time:

$$TFT = \sum_{i=1}^{n} C(\pi_i, m) \tag{7}$$

3. Block-Based Artificial Chromosome Rapid Non-Dominated Sorting Genetic Algorithm Ⅱ

This section includes population initialization, construction of artificial chromosomes, chromosomal recombination and retention of dominant solutions. P-ACNSGA-II flow chart shown in Fig. 1.

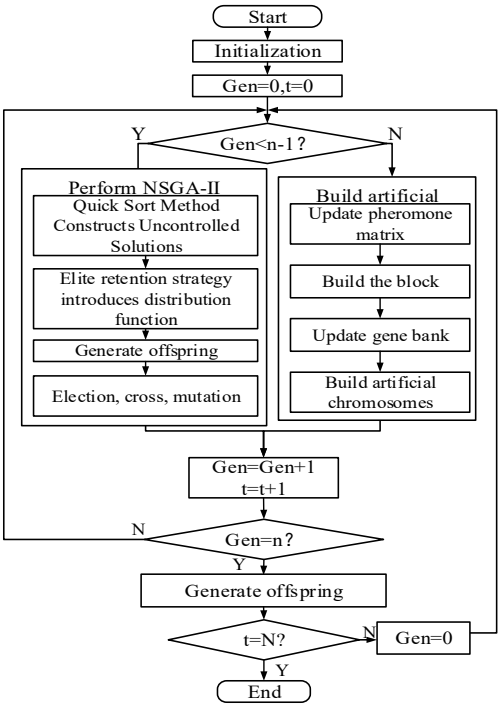


Fig. 1 p-ACNSGA-II flow chart

3.1 Population Initialization

3.1.1 Application Improved the Opposition-Based Learning Population Initialization

The main idea of opposition-based learning (OBL) (adopted from Lin et al. 2016b) is to consider its opposite sequence simultaneously to obtain the optimal solution in the current candidate solution. Let $X=(x_1,x_2,\ldots, x_n)$ be a solution in an N-dimensional space, where $x_i \in R$ and $x_i \in [l_i,u_i]$, l_i and u_i are the upper and lower bounds of the search space, $\forall i \in \{1,2,\ldots,n\}$, then its oppositional solution is $X'=(x_1',x_2',\ldots,x_n')$, where $x_i'=l_i+u_i-x_i$. OBL optimization process is as follows:

Step 1: Generate an initial solution $X=(x_1,x_2,\ldots, x_n)$ in the n-dimensional search space and its opposition-based solution $X'=(x_1',x_2',\ldots,x_n')$.

Step 2: Evaluate the fitness of two solutions. Find the fitness $f(x)$ of the initial solution and the fitness $f(x')$ of the opposition-based solution.

Step 3: If $f(x')\geq f(x)$ replaces x with x'; otherwise, continue using x.

Get better results by evaluating both solutions simultaneously.

In this study, the opposition-based learning method is improved to reduce the loss of good solution. The specific process is first to generate an initial solution of N (N is a given initial population size) by random method and find its opposition-based solution for each initial solution; and then the two solutions are mixed together to form a scale of 2N Population, calculate the fitness of each solution, and the solution according to the size of the fitness in descending order; the last choice of the top N adaptive fitness evolution of the initial population.

3.2 Improved Non-Dominance Sorting Genetic Algorithm Ⅱ

The algorithm measures the crowding degree of individuals by calculating the crowding distance so that the boundary non-dominated solution can be effectively preserved. This study improves on the two aspects of non-dominated sorting process and elite retention strategy respectively.

3.2.1 Elite Retention Strategy

NSGA-II classifies the population by non-dominated sorting to form a non-dominated hierarchical surface $F=\{F_1,F_2,\ldots\ldots F_i\}$, which is ranked in ascending order to select excellent populations as offspring. However, the traditional NSGA-II does not limit the number of individuals in each level,

which will lead to the majority of non-dominated solutions in the same level, and thus far away from the real Pareto optimal surface. For the reason, this study will refer to the distribution function (As in Equation (8)) (adopted from Gao,2012a). Each layer of Pareto surface by adding the right amount of non-elite solution or reduce the amount of elite solution to ensure the diversity of populations.

$$n_i = |F_i| \times r_i \tag{8}$$

Where i represents the serial number of Pareto surface; n_i represents the number of individuals selected on the Pareto surface F_i, $|F_i|$ represents the total number of individuals on the Pareto surface F_i, r_i represents a coefficient on the i th Pareto surface, and $1 > r_i > 0$.

The process is as follows:

Step 1: Combine the parent (P_t) and the offspring (Q_t), each having a population size of N, to form a population R_t and its size is $2N$;

Step 2: Non-dominated sorting of population R_t to form Pareto grade surface, $F = \{F_1, F_2, \ldots\ldots F_i\}$;

Step 3: According to the Equation (8), calculating n_i ,adding the first n_i individuals in F_i to P_{t+1} .If $N - \sum n_{i-1} \le n_i$, calculate the crowding distance of individuals not selected Pareto surfaces, and select the individuals with larger crowding distances to put in P_{t+1} until the number of individuals in P_{t+1} equals N.

3.3 A Block Mining and Artificial Chromosome Generation Approach

3.3.1 Mining Block

In genetic algorithms, most chromosomes of progenies that have been iterated for several generations carry better information, which has similarities and these similarities have certain reliability in large-scale populations (adopted from Chang et al. 2013c). In p-ACNSGA-II, the ACO pheromone concentration is used to identify better similar sequences in the chromosome and to excise them by means of blocks.

(1). Establish the pheromone matrix

In this study, we use the dependent pheromone matrix and the position-pheromone matrix to record the path information of ants. The pheromone matrix records the pheromone concentration at each node. The dependent pheromone matrix records the pheromone concentration of the path between two nodes .

Step 1: Initialize the pheromone matrix:

For each turn of the genetic algorithm, the (n-1)th generation is descended according to its fitness, and an optimal chromosome is selected for matrix initialization. Fig. 2 shows the initial dependent pheromone matrix generation process. The initial pheromone between two workpieces is expressed as (9).

$$\tau_0 = \frac{1}{L} \text{ (} L \text{ represents makespan)} \tag{9}$$

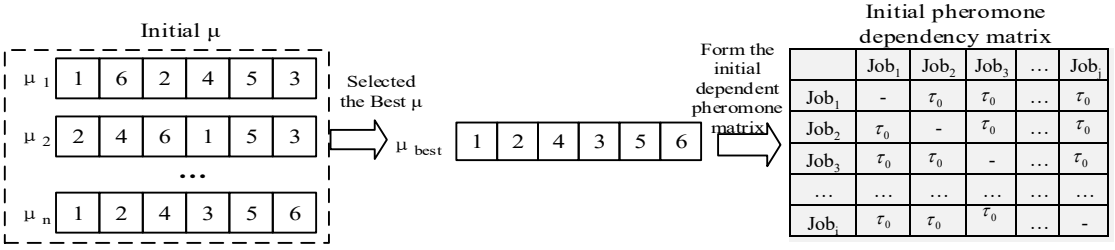


Fig.2 Initial Dependent Pheromone Matrix

Step 2: Update pheromone matrix:

The first 30% of the chromosomes were selected to update the pheromone matrix in the ranked (n-1) generation. Fig.3 shows the updating process of the dependent pheromone matrix. The updating

formula is as shown in Equation(10). τ_{ij} represents the pheromone concentration of each pair of workpieces (i, j), $\tau_{ij}(t+1)$ represents the updated τ_{ij}, ρ represents the evaporation rate of pheromone (adopted from Dorigo et al. 1997).

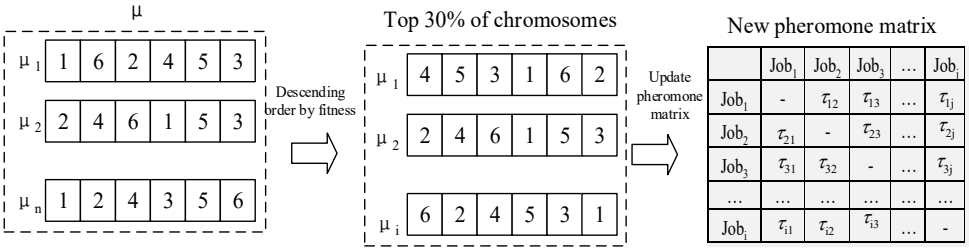


Fig. 3 Update Dependent Pheromone Matrix

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \rho \Delta \tau_{ij}(t+1)$$
$$\Delta \tau_{ij}(t+1) = \begin{cases} 1/L & \text{If the ants go through the path } (i, j) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Similarly, the updated position pheromone matrix is shown in Fig. 4.

	S_1	S_2	S_3	…	S_m
J_1	τ'_{11}	τ'_{12}	τ'_{13}	…	τ'_{1m}
J_2	τ'_{21}	τ'_{22}	τ'_{23}	…	τ'_{2m}
J_3	τ'_{31}	τ'_{32}	τ'_{33}	…	τ'_{3m}
…	…	…	…	…	…
J_i	τ'_{i1}	τ'_{i2}	τ'_{i3}	…	τ'_{im}

Fig. 4 Updated Location Pheromone Matrix

(2). Building the block

First determine the minimum length of the block, and then randomly select the starting position, and then select the process for the starting position. Within the minimum block length, the process at the start position is selected based on the information in the position pheromone matrix. Processes at other locations use the combined information of the location and the dependent pheromone matrix. The process is screened by the roulette algorithm (RWS). The equation of position probability matrix is as Equation (11), the Equation of dependent probability matrix is as Equation (12) and the Equation of combined probability as Equation (13). In Fig.5, assuming that P'11> P'21> P'31> ...> P'i1, the workpiece J1 is selected to be placed in the position S1. In Fig.6, it is assumed that CP2 is the largest and J2 is selected as the position S2. Choosing randomly when two larger values appear.

$$P'_{im} = \frac{\tau'_{im}}{\sum_{i \in uns} \tau'_{im}} \qquad i, m = 1, ..., n \quad (11)$$

$$P_{ij} = \frac{\tau_{ij}}{\sum_{i \in uns} \tau_{ij}} \qquad i, j = 1, ..., n \quad (12)$$

$$CP_i = (W \times P_{ij}) + (W' \times P'_{im}) \quad i, j, m = 1, ..., n \quad (13)$$

i represents the part number. j represents the last part number connected to i. m represents the position of the gene on the artificial chromosome. n represents the length of the artificial chromosome. N represents the population size. P'_{mi} represents the probability of the position of the workpiece i and the position m in the position matrix. P_{ij} represents the probability that the workpiece j is connected to the workpiece i in the dependent matrix. CP_i represents the combined probability of the workpiece i. W' and W respectively represent the weight values of the position matrix and the dependent matrix. As

the evolutionary algebra increases, the weight of the dependent matrix decreases from 0.7 to 0.3, and the position matrix instead (adopted from Pei et al. 2017k).

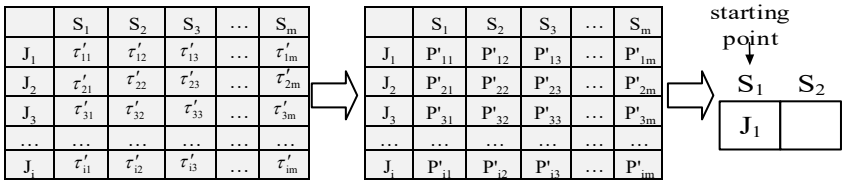


Fig. 5 Starting Position Process Selection

For the process selection of the position outside the minimum length, a minimum combined probability threshold [adopted from Pei et al.2017l] is set as the screening condition, as shown in Figure 7. When the block contains more parts, the lower the overall probability, the greater the probability of the combination of errors, the block threshold will ensure the quality of the block. Mined blocks are stored in the block database.

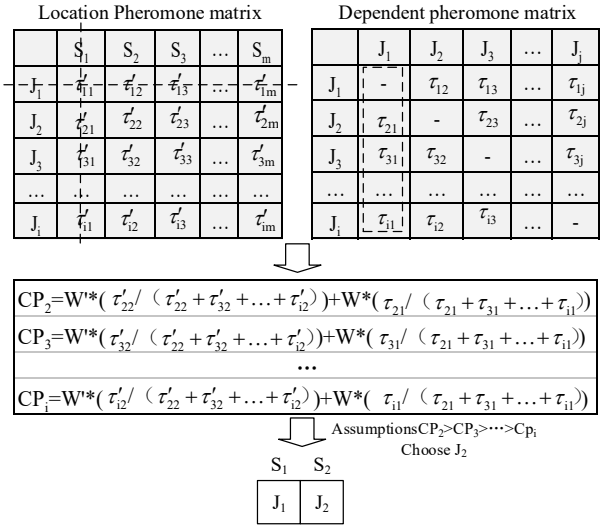


Fig.6 Block Minimum Length of other Parts of the Workpiece Selection

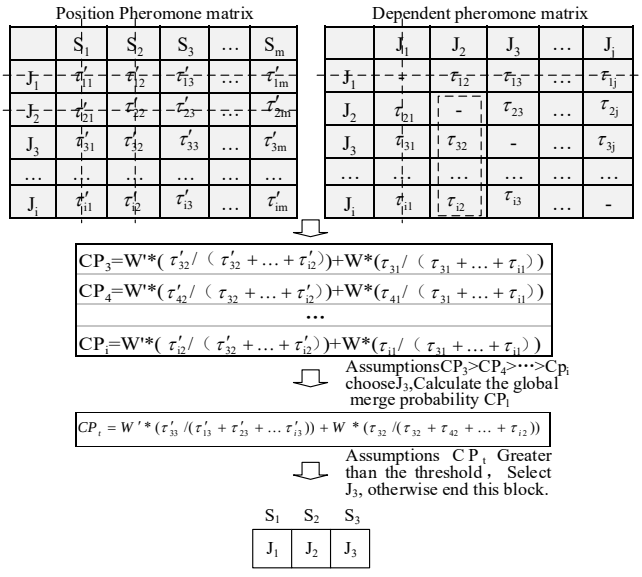


Fig.7 Block outside the Minimum Length of the Workpiece Selection

(3). Block competition

Block competition (adopted from Zhang et al.2017m) is to compare the process and position of the blocks in the block database. If there are repeated processes between the blocks or the covered positions overlap, the repeated blocks are compared by the average probability.Those with a higher average probability are retained in the block repository, while the smaller ones are deleted.The average probability calculation method of the block is as shown in Equation (14).

$$p_{B^i}^{AVG} = \frac{p_{B_1^i}^{dom} + \sum_{l=2}^{n} CP_{B_l^i}}{n} \quad l = 1,\ldots,n \tag{14}$$

i represents the block number. l represents the l th position of the block. n represents the length of the block. B_1^i represents the i th workpiece of the i th block. $CP_{B_l^i}$ represents the merge of the l th workpiece of the i th block Probability.

3.3.2 Combination of Chromosomes

This study uses the blocks reserved in the block database to combine the best chromosomes to improve the solution quality and convergence speed of the algorithm. All blocks in the block database are copied to a corresponding position on the blank chromosome of a certain length. The rest of the procedure was selected using the RWS method at the empty position in the chromosome.

3.4 Preserving the Dominant Solution

The (n-1)th generation μ_i and the generated artificial chromosome Ci are put in the selection pool, and the excellent chromosomes are selected as the offspring to enter the next round of evolution using the binary competition method (adopted from Chang et al.2014a).The specific process is as follows: We randomly select two chromosomes from the selection pool, Compare the degree of fitness, select the chromosome with higher fitness to put into the chromosome library, and put the chromosome with less fitness into the selection pool to continue the screening. The above steps are repeated until the number of chromosomes in the chromosome bank satisfies the set population size.

4. Experimental Results

The algorithm proposed in this article is written in C++ language. The operating environment of the program is a computer with Intel (R) Core (TM) i5-4005U CPU @ 3.40GHz and memory of 4.0G. To test the performance of the p-ACNSGA-II algorithm, two series of Taillard and Reeve examples were chosen for testing, and the test results were compared with the well-known NSGA-II algorithm. In order to make the test results comparability, the two algorithms have the same parameter values in the same category of parameters, the population size is 100 and the execution algebra is 200. Experiments of Anurag et al.(2014b) showing, NSGA-II can achieve better results when the crossover rate is 0.70 and the mutation rate is 0.10. The same crossover rate and mutation rate are used in this study.

Tab.1 shows results of p-ACNSGA-II and NSGA-II about the Reeve instances and Tab.2 shows results of p-ACNSGA-II and NSGA-II about the Taillard instances. n * m means that there are n jobs and m machines. It can be seen from tables that for both Reeve instances and Taillard instances, NSGA-II is superior to p-ACNSGA-II in makespan for individual instances, but for most of the two goals of makespan and total flow time p-ACNSGA-II has better test results than NSGA-II in both makespan and total flow time. It is verified that the p-ACNSGA-II algorithm performs better than the NSGA II algorithm.

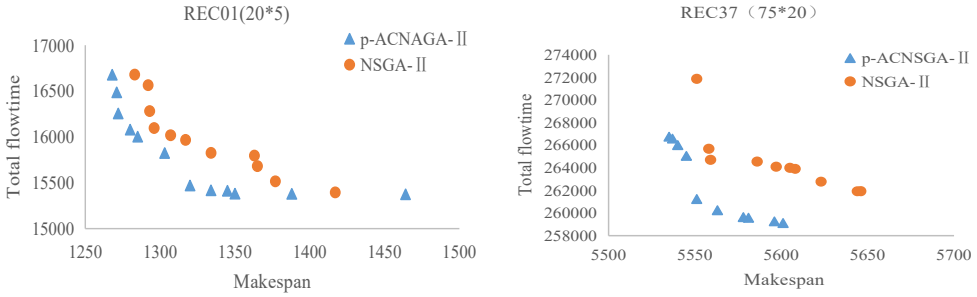
Table. 1 Comparison of Reeve Instance Test Results(a)

REC01(20*5)				REC37(75*20)			
p-ACNSGA-II		NSGA-II		p-ACNSGA-II		NSGA-II	
Makespan	Total flowtime	Makespan	Total flowtime	Makespan	Total flowtime	Makespan	Total flowtime
1268	16871	1283	16680	5535	266753	5551	271868
1271	16487	1292	16564	5537	266582	5558	265682
1272	16356	1293	16282	5540	266007	5559	264709
1273	16197	1296	16097	5545	265056	5586	264551
1285	16142	1307	16019	5551	261239	5597	264101
1303	15825	1317	15967	5563	260244	5605	264002
1382	15470	1334	15826	5578	259640	5608	263909
1401	15418	1363	15796	5581	259578	5623	262776
1406	15413	1365	15682	5596	259263	5644	261926
1415	15382	1377	15516	5601	259118	5646	261928

Table. 2 Comparison of Taillard Instance Test Results (a)

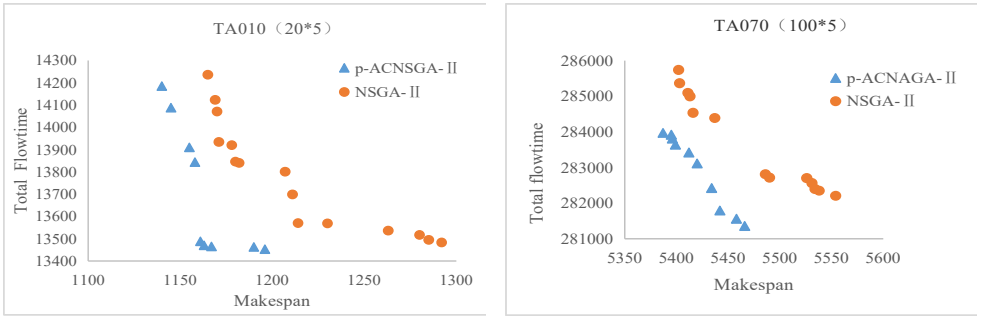
TA010(20*5)				TA070(100*5)			
p-ACNSGA-II		NSGA-II		p-ACNSGA-II		NSGA-II	
Makespan	Total flowtime	Makespan	Total flowtime	Makespan	Total flowtime	Makespan	Total flowtime
1140	14185	1165	14236	5387	283970	5402	285737
1145	14088	1169	14123	5395	283920	5403	285365
1155	13910	1170	14071	5396	283811	5411	285094
1158	13844	1171	13934	5399	283642	5413	284996
1161	13488	1178	13920	5412	283422	5416	284534
1163	13470	1180	13846	5420	283113	5437	284389
1167	13465	1182	13840	5434	282424	5486	282812
1190	13463	1207	13801	5442	281795	5490	282716
1196	13453	1211	13699	5458	281564	5526	282703
-	-	1214	13570	5466	281362	5531	282564

Fig.8 and Fig.9 show the Pareto optimal frontier comparison of p-ACNSGA-II and NSGA-II. It can be seen from the figure that p-ACNSGA-II has better convergence than NSGA-II for all Taillard instances and Reeve instances, indicating that p-ACNSGA-II has more search space than NSGA-II.



(a) (b)

Fig.8 Pareto Optimal Frontier for Reeve Series Instances



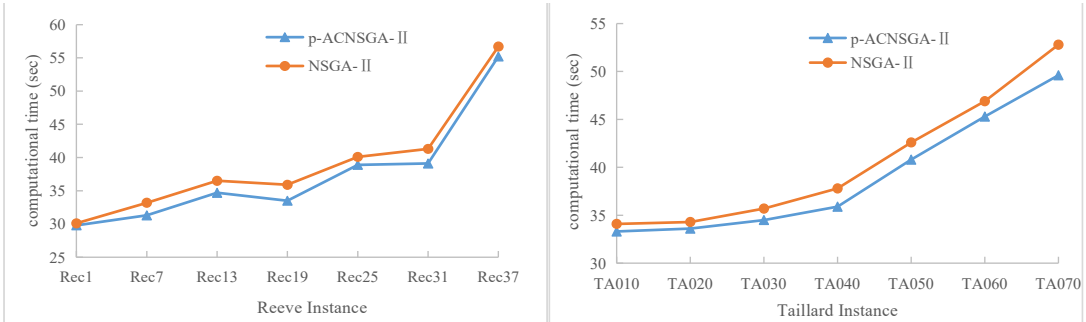
(a) (b)

Fig. 9 Pareto Optimal Frontier for Taillard Series Instances

Tab. 3 shows the average computational time for each operation in each instance of the table, and Fig. 10 shows the computational time line chart. As can be seen from the table and the figure, although both p-ACNSGA-II and NSGA-II use the same non-dominated sorting technique, p-ACNSGA-II took less time than NSGA-II in almost all cases, indicating the validity of the block.

Table.3 Comparison of Average Calculation Time

Instance	Number jobs(n) and machines(m) (n × m)	Average computational time (sec)	
		p-ACNSGA-II	NSGA-II
Rec01	20*5	29.8	30.2
Rec07	20*10	31.3	33.3
Rec13	20*15	34.7	36.4
Rec19	30*10	33.5	35.8
Rec25	30*15	38.9	40.2
Rec31	50*10	39.1	40.8
Rec37	75*20	55.2	56.5
TA010	20*5	33.3	34.7
TA020	20*10	33.6	34.3
TA030	20*20	34.5	35.7
TA040	50*5	35.9	37.4
TA050	50*10	40.8	41.9
TA060	50*20	45.3	46.7
TA070	100*5	49.6	52.6



(a) (b)

Fig.10 Linear Comparison of Calculation Time

5. Conclusion

In order to minimize the maximum makespan and total flow time of the permutation flow shop, a block-based artificial chromosome non-dominated sorting genetic algorithm (p-ACNSGA-Ⅱ) is proposed. p-ACNSGA-II combines genetic algorithm with ant colony algorithm, improving the speed

of evolution and reconciliation. The test results are compared with NSGA-II by testing Reeve and Taillard benchmark examples. From the comparison results, it can be seen that p-ACNSGA-Ⅱ has better solving performance and better Pareto optimal frontier, Shorter than NSGA-II. It can be seen that p-ACNSGA-Ⅱ has better performance in solving multi-objective permutation flow shop scheduling problem. Future research can solve other multi-objective combinatorial problems such as traveling salesman problem and vehicle routing problem by further improving the algorithm.

References

[1]. HUANG Xia, Ye Chunming, Cao Lei. Optimization of Chaos Weed Based on Multi-objective Displacement Flow Shop Scheduling [J]. Systems Engineering Theory and Applications, 2017a,37 (01): 253-262.

[2]. JinDeng,LingWang.A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem[J].Swarm and Evolutionary Computation Volume 32,February 2017d, Pages 121-131.

[3]. Achmad P.Rifai,Huu-ThoNguyen,Siti Zawiah MdDawal.Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling[J].Applied Soft Computing.Volume 40, March 2016a, Pages 42-57.

[4]. Pei-Chann Chang, Wei-Huang, Jheng -Long Wu, et al. A block mining and re-combination enhanced genetic algorithm for the permutation flowshop scheduling problem[J].Int. J. Production Economics 141 (2013b) 45–55.

[5]. Jian Lin.A hybrid discrete biogeography-based optimization for the permutation flow shop scheduling problem[J].International Journal of Production Research, (2016b)54:16, 4805-4814.

[6]. Pei-Chann Chang , Wei-Hsiu Huang. A block mining and re-combination enhanced genetic algorithm for the permutation flowshop scheduling problem[J]. Int. J. Production Economics 141 (2013c) 45–55.

[7]. Dorigo, M., Gambardella, L.M., 1997. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolution- ary Computation 1, 53–66.

[8]. Chen Huifen.Coupled learning based sub-population evolutionary algorithm for solving multi-objective scheduling problem [D]. Tianjin University of Technology, 2017j.

[9]. PEI Xiao-bing, CHEN Hui-fen, ZHANG Bai-stack eatl.Modified BVEDA for Multi-objective Scheduling Problem [J]. Journal of Shandong University (Engineering and Technology), 2017k,47 (04): 25-30.

[10]. ZHANG Min, WANG Yang, FANG Kan.An improved block evolutionary algorithm for solving the problem of displacement flow shop [J]. Computer Integrated Manufacturing Systems: 1-19. http://kns.cnki.net/kcms/detail/ 11.5946.TP.20170924.1531.010.html.

[11]. Pei-ChannChang,Meng H.C.A block based estimation of distribution algorithm using bivariate model for scheduling problems.SoftComput (2014a) 18:1177–1188.

[12]. Anurag Tiwari, Pei-Chann Chang, M.K. Tiwari, et al. A Pareto block-based estimation and distribution algorithm for multi-objective permutation flow shop scheduling problem [J].Production Research, 14 Jul 2014b.