

## Elite Particle Swarm Optimization Algorithm for Solving the Bi-Criteria No-wait Flexible Flow Shop Problem

Yongbin Qin<sup>1,2\*</sup> and Haiyue Zhang<sup>2</sup>

<sup>1</sup>Guizhou Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, P.R.China

<sup>2</sup>College of Computer Science and Technology, Guizhou University, Guiyang, 550025, P.R.China  
ybqin@foxmail.com

### Abstract

*The thesis mainly studies bi-criteria no-wait flexible flow shop problem, whose optimization objective is to minimize the maximum completion time and the maximum delay time. This problem is NP hard, yet enjoying important theoretical research value, thereby this thesis proposes elite particle swarm optimization (EPSO) to solve bi-criteria no-wait flexible flow shop problem. EPSO algorithm applies five modified heuristic algorithms and random methods to generating initial population. Moreover, for the particle personal best, this thesis puts forward elite crossover algorithm, which retains continuous fragments of the identical workpieces among excellent individuals, avoiding the destruction of good continuity between solutions of workpieces. In addition, in order to avoid algorithm into local optimum, this thesis raises double insertion disturbance algorithm to help particles jump out the local optimal state and expand the feasible search range. For the purpose of effectively evaluating algorithm quality, there is a comparison among EPSO algorithm, PSO algorithm and ICA algorithm in simulation experiment that is respectively aimed at small-scale problem and large-scale scheduling problem, the results of which show that the proposed EPSO algorithm, due to better validity and accuracy, is superior to the PSO algorithm and ICA algorithm.*

**Keywords:** flexible flow shop; bi-criteria; no-wait; elite crossover

### 1. Introduction

Flexible flow shop problem is a crucial branch of scheduling issue [1] and a frequent model in actual industrial production. General flow shop problem involves allocation of fixed processing machine tools, while the flexible flow shop problem [2] allows the application of parallel machines, during which, the workpiece in each working procedure can arbitrarily select parallel machine to process. In 1971, Arthanari and Ramamurthy [3] for the first time proposed flexible flow shop scheduling model. After then, researchers came up with the branch and bound method to solve this problem [4]. This algorithm can get the optimal solution but only apply to solving the small-scale problem. However, when the size of problem is large enough, it requires exponential time costs, thus can not be used in actual production. Since then, researchers proposed heuristic algorithm to tackle the problem [5]. For instance, Mainieri G. B. and Ronconi D. P. [6] applied three allocation rules, priority of the earliest due date, priority of the latest due date and priority of the maximum time delay in order to bring initial sort. In this thesis then, a feasible solution is found by using forward list scheduling algorithm (the traditional scheduling algorithm and new scheduling algorithm considering the idle time), backward list scheduling algorithm and bottleneck center scheduling algorithm. The experiment results

indicate that the new scheduling algorithm which combines the priority of maximum delay time outperforms other optimization scheduling algorithms. Nevertheless, although the heuristic algorithm is faster, it is difficult to solve large-scale problems and guarantee quality of solution. In recent years, the smart optimization algorithm is gradually on the rise, which can be widely applicable since it is accessible to better feasible solution within an acceptable time. For example, Jitti Jungwattanakit, Reodecha M., Chaovalitwongse P. etc. [7] put forward a genetic algorithm to resolve the flexible flow shop problem with loading time, the application of which is to use polynomial heuristic optimization method based on the transfer of workpiece, so as to optimize feasible solutions of algorithm. Zandieh. M., Ghomi SMTF, Hussein SMM [8] studied the hybrid flow shop scheduling problem constrained by loading time with dependent sequence; moreover, on the basis of immune algorithm, they utilized the accelerating mechanism to recombine the current optimal solutions of algorithm to search the approximate optimal solution, which is able to accelerate the convergence rate of algorithm, and meanwhile mitigate the control advantage of the optimal solution to prevent the algorithm into a local optimum.

No-wait flow shop scheduling issue [9-11] is extremely significant in flow shop scheduling field. For example, in steel-making process, iron and steel enterprises require liquid steel should be finished continuously in the whole process or satisfy no-wait requirements in order to guarantee the mode of production of hot delivery and hot charging; chemical production is also needed to meet no-wait requirements, because the emergence of time gaps during process may lead to serious consequences such as chemical deterioration. Researchers have demonstrated that no-wait flow shop scheduling problem with two or more processing machines is strongly NP hard [12]. Therefore, no-wait flow shop scheduling issue shares significantly applicable value and research value, gaining increasingly attention from researches. Khalili M. and Naderi B. [13] treated the invasive weed optimization technique as a global search algorithm, and integrated variable neighborhood search algorithm and simulated return algorithm together to achieve local search, so the integration of these three meta heuristic algorithms makes the search process more effective and diversified. Liu Zhixin etc. proposed the least deviation algorithm (LD) to solve no-wait flexible flow shop problem, and Liu's algorithm is easier than those of Johnson and List in terms of algorithm time. Moradinasab N., Shafaei R., Rabiee M. [15] and others put forward an adaptive imperialist competitive algorithm and a genetic algorithm for solving no-wait flexible flow shop problem; besides, their experiment results demonstrate that these two algorithms are better than ant colony optimization algorithm. For no-wait flexible flow shop problem with two stages, M. Rabiee [16] came up with a hybrid algorithm (HA) based on imperialist competitive algorithm. By means of variable neighborhood search algorithm (VNS) to expand search scope, HA algorithm obtains new individuals through integrating mutation and crossover operation of genetic algorithm (GA), and determines whether to receive new individuals by simulating annealing method. The experiment results prove that HA enjoys better performance than ICA, SA, VNS, GA, ACO and many other algorithms.

Most researches of flexible flow shop problem [17-19] focus on optimization of a single objective, but the actual scheduling problem is characterized multiple objectives. In recent years, some scholars have studied multi-criteria flow shop scheduling problem. For instance, Danneberg[20] ranked the workpieces to be sorted which then went through batch processing and presented workpieces sort; they also put forward various structural type and iterative algorithm with maximum completion time and job completion time as objective function. Allahverdi and Aldowisan[21] brought forward two heuristic algorithms, hybrid simulated annealing algorithm and hybrid genetic algorithm, for tackling no-wait flow shop problem whose goals are maximum completion time and maximum delay time; at the meanwhile, they de-

defined the dominance relationships (DR) between feasible solutions. A large number of experimental data prove that the heuristic algorithm proposed in this thesis enjoys higher optimization performance than the best heuristic algorithms at that time.

Bi-criteria no-wait flexible flow shop problem with multiple stages is one of the most crucial problems in the actual production process, but as of now, researches in this filed have had a lot to be desired. Therefore, with the goal of minimizing the maximum completion time and the maximum delay time, the thesis proposes elite particle swarm optimization algorithm (EPSO) to tackle no-wait flexible flow shop problem.

## 2. Problem Description

### 2.1 No-wait Flexible Flow Shop Problem

For no-wait flexible flow shop problem,  $n$  workpieces  $\{J_1, J_2, \dots, J_n\}$  are needed to be processed, all workpieces are required to be processed through  $m$  working procedures  $\{S_1, S_2, \dots, S_m\}$ , and the processing paths of all workpieces are the same. Among them, working procedure  $i$  includes  $S_i$  numbers of parallel machine, and all workpieces can be processed in any parallel machine; the process time of the workpiece  $J_j$  on the  $k$  machine in the  $i$  working procedure is  $t_{ij}^k$ , and the due date of workpiece  $j$  is  $d_j$ . No-wait flexible flow shop problem must satisfy the following constraints:

Each process should at least has one processing machine, and parallel machines in a certain process must not less than two;

Once the workpiece is started to be processed it cannot be interrupted, namely the workpiece must be continuously completed all stages of processing, and processing period forbids may interruption or stealing;

The release time of all workpieces is zero, namely, all workpieces can be processed in time zero.

### 2.2 Bi-criteria Optimization

The first optimal criteria  $f_1$  is the maximum completion time. To assume  $C_j$  as the completion time of workpiece  $j$ , the maximum completion time  $C_{\max} = \max(C_j)$ , then  $f_1 = C_{\max}$ .

The second optimal criteria is the maximum delay time  $f_2$ . To set  $T_j = \max(0, C_j - d_j)$  as the delay time of workpiece  $j$ , among which,  $d_j$  is the delivery time (or due date) of workpiece  $j$ , then  $f_2 = T_{\max} = \max(T_j)$ .

The optimal criteria of the entire problem is  $\min f = (f_1, f_2)$ .

In the thesis, the objective functions *makespan* and the maximum delay time are set as weight  $\omega_1$  and  $1 - \omega_1$  respectively, among which,  $0 \leq \omega_1 \leq 1$  and  $\omega_1$  increases with 0.1 step, then  $\min f = \omega_1 \times f_1 + (1 - \omega_1) \times f_2$ .

## 3. Elite Particle Swarm Optimization Algorithm

Since Particle swarm optimization algorithm is easy to understand, convenient in realizing and strong in searching, this algorithm occupies a crucial position in many optimization problems. On the basis of particle swarm algorithm, the thesis proposes the elite particle swarm optimization algorithm (EPSO) to solve bi-criteria no-wait flexible flow shop problem. Firstly, EPSO algorithm makes use of discrete encoding method. In the initial phase of the algorithm, five heuristic algorithms are

applied to generating five better initial solutions, and other particle information is brought randomly while satisfying constraints. In addition, particle constantly updates its position and velocity information according to the flight experience of its own and groups, and closes to the optimal position; if the personal best generates many different current optimal solutions, better particles are produced in personal best set through elite crossover algorithm. This thesis also introduces the double insertion disturbance algorithm to avoid the feasible solution into local optimum.

Based on the above explanations, the EPSO algorithm proposed in this thesis is described as follows:

---

**EPSO algorithm**

**Input:** Population size  $PS$ , job matrix  $PT$  (the number of jobs is  $n$ ), parallel machine information  $SN$ , iterations  $TotalIte$ , probability of Elite crossover algorithm  $p_e$ , probability of Double insertion disturbance algorithm  $p_{di}$

**Output:** Optimal solution set  $Sch$

---

1.  $s[PS][n] \leftarrow 0$ ; //define initial population
  2. **for**  $i = 1:5$
  3.   Use modified five heuristic algorithms to generate five initial solutions;
  4.   Record individual best position set  $p_i$  of every particle;
  5. **end**
  6. **for**  $i = 6:PS$
  7.    $s[i] = random$ ; // Generate other initial populations randomly;
  8.   Record individual best position set  $p_i$  of every particle;
  9. **end**
  10. **for**  $j = 1:TotalIte$
  11.   **for**  $i = 1:PS$
  12.     Update the location and speed information of particles;
  13.     **if**  $rand < p_e$
  14.        $p_i \leftarrow$  Update the individual best position by using Elite crossover algorithm ;
  15.     **end**
  16.   **end**
  17.    $g \leftarrow$  Global best position;
  18.   **if**  $rand < p_{di}$
  19.      $g \leftarrow$  Update global best position by using double insertion perturbation algorithm;
  20.   **end**
  21.   Update inertia weight  $\omega$  ;
  22. **end**
  23. **return**  $Sch = g$  ;
- 

In the following part of the thesis, the proposed algorithm is presented in details, including the encoding method, population initialization, particle position updating, velocity updating method, crossover operator and the method of avoiding local optimum.

### 3.1 Five Heuristic Algorithms for Generating Initial Population

Since initial population of algorithm exerts a great impact on solving velocity of algorithm, the thesis utilizes five optimal particles in the initial population of elite particle swarm optimization algorithm generated by five heuristic algorithms, which includes three heuristic algorithms through the modification of CDS, NEH, SDH as well as the algorithm of the earliest due date priority and the algorithm of single objective hybrid.

For the purpose of operation and simplicity, the first step is to calculate the following three pieces of information when workpiece  $j$  is in the  $i$  process, namely, the

maximum process time  $MaxT_{i,j} = \max_{k \in S_i} (t_{i,j}^k)$  ; the minimum process time  $MinT_{i,j} = \min_{k \in S_i} (t_{i,j}^k)$  ; and the average process time  $AverT_{i,j} = mean_{k \in S_i} (t_{i,j}^k)$ .

The original heuristic algorithm is suitable for the traditional flow shop scheduling problem rather than the flexible flow shop scheduling problem, so the flexible flow shop problem with  $m$  phases is regarded as the permutation flow shop scheduling problem with  $m$  stages, and the maximum process time, the minimum process time or the average process time of workpiece in each stage is treated as the process time in the corresponding stage of the permutation flow shop scheduling problem. These five heuristic algorithms are briefly described as follows.

1) MCDS (Modified CDS algorithm): to apply  $MaxT_{i,j}$ ,  $MinT_{i,j}$  and  $AverT_{i,j}$  as the respective process time of workpiece  $j$  in working procedure  $i$ ; to divide stage  $m$  into  $m-1$  group  $F_2 | prmu | C_{max}$ , get the optimal scheduling  $\pi_1, \pi_2$  and  $\pi_3$  by Johnson algorithm and then define an initial solution by selecting target value in these three sorts.

2) MNEH (Modified NEH algorithm): to use  $MaxT_{i,j}$ ,  $MinT_{i,j}$  and  $AverT_{i,j}$  as the respective process time of workpiece  $j$  in working procedure  $i$ ; to obtain three initial sorts by NEH standard algorithm and select the optimal scheduling in these three sorts.

3) MSDH (Modified SDH algorithm): In the algorithm,  $MaxT_{i,j}$ ,  $MinT_{i,j}$  and  $AverT_{i,j}$  are treated as the process time of workpiece. Then three sorts can be obtained by NEH algorithm, and the feasible solution is selected through the optimal scheduling.

4) EDD: The workpiece sort is ranked according to the ascending order of due date of the workpiece to be processed.

5) SOH (Single Objective Hybrid algorithm): Solutions should be got respectively with the aforesaid NEH algorithm, applying *makespan* and the maximum delay time as the optimal scheduling  $\pi_1, \pi_2$  of optimal objectives, and combining  $\pi_1, \pi_2$  as the optimal scheduling of sort  $\pi$ . Among them, the workpiece sorts with the same positions in  $\pi_1, \pi_2$  should be copied to the corresponding positions in  $\pi$ , and the rest sorts are copied after random number selects a corresponding position information in  $\pi_1, \pi_2$ .

### 3.2 Encoding

In this thesis, discrete particle information is known as workpiece sort, particle dimension as number of workpieces to be sorted, and value of particle in each dimension as workpiece number. As shown in Table1, in a scheduling problem with seven workpieces to be sorted, the workpiece sort through particle information is  $\pi = \{J_2, J_3, J_5, J_6, J_1, J_4, J_7\}$ .

**Table 1. Coding**

Particle information	2	3	5	6	1	4	7
Job order	2	3	5	6	1	4	7

### 3.3 Population Initialization

In particle swarm optimization algorithm, each particle includes two pieces of basic information, particle position and velocity; therefore, the algorithm involves the position information of the five particles in initial populations generated by MCDS, MNEH, MSDH, EDD and SOH algorithms; the position information of the rest particles and the velocity information of all particles are produced randomly. In order to calculate the target value of all particles, particle personal best and global

best, the target value of particle  $i$  is marked as  $T_{gi}$ , personal best  $p_i$  and global best  $g_{best}$ .

### 3.4 Particle Position and Velocity Updating

Flexible flow shop problem shares its special characteristics, so particle position and velocity information require detection of legitimacy, yet the upgrading method of basic particle swarm is not suitable any more. Consequently, it is essential to modify the formula of particle position and velocity updating, and the modified formula of velocity updating is showed as follows:

$$v_i^{t+1} = \omega v_i^t \times c_1 p_i \times c_2 g_{best} \quad (1)$$

Among it,  $c_1 \cdot c_2 = 0, c_1 + c_2 = 1$ .  $v_i^{t+1}$  and  $v_i^t$  represent the velocity information of particle  $i$  at the time  $t+1$  and time  $t$ .  $\omega$  is inertia weight,  $p_i$  the current personal best of particle  $i$ ,  $g_{best}$  the global best, and  $\times$  the updating operation. The optimal solution of bi-criteria optimization problem often is a set, thereby personal best and global best may be a set, and then the updating operation in personal best and global best randomly selects an optimal solution to calculate. In particle swarm optimization algorithm, tendency to self-cognition at the initial stage of particle and tendency to social cognition can effectively prevent the algorithm from local optimum. Thereby, at the initial stage, the algorithm chooses  $c_1 = 1, c_2 = 0$ , velocity information and personal best (or self-cognition) with a larger probability to go through updating operation; while at the latter stage, the algorithm selects  $c_1 = 0, c_2 = 1$ , velocity information and global best (or social cognition) to experience updating operation.  $\omega$  applies the strategy of decreasing concave function to updating. The updating formula of the corresponding particle position information after modifying is:

$$x_i^{t+1} = x_i^t \times v_i^{t+1} \quad (2)$$

$v_i^{t+1}$  and  $x_i^{t+1}$  stand for the velocity information and position information of particle  $i$  at the time  $t+1$ . The updating position information of particle  $i$  at the time  $t$  is the result of the updating information between the position information at the time  $t$  and the velocity information at the time  $t+1$ .

In order to guarantee the legitimacy of velocity information and particle position after updating, the updating operation is defined as follows: first, it should compare the information of the two particles in the calculating operation and keep the work-piece order with the same information at the same position unchanged. Secondly, it is essential to randomly produce the selective information  $\Delta$ , which shares the same dimension with particle and whose domain is in  $\{0,1\}$ . Apart from the aforesaid fixed particle information, other positions should be updated in accordance with  $\Delta$ ; if the corresponding information of  $\Delta$  is 1, the information of particle  $x_1$  should be replaced with the information at the position  $v_1$ ; if the aforesaid information is 0, the operation should be omitted. The rest position information is decided by the workpieces to be arranged in  $x_1$ .

**E.g.1:** There are seven workpieces to be processed  $\{J_1, J_2, J_3, J_4, J_5, J_6, J_7\}$  in a flexible flow shop problem. Showed as Figure 1,  $x_1$  is the position information of a particle,  $v_1$  the velocity information of the particle and  $\Delta$  the randomly selective information. And  $x_2$  serves as the new particle information brought by updating operation  $x_1 \times v_1$ .

### 3.5 Elite Crossover Algorithm

Elite individual boasts the individual enjoying the highest target value searched by algorithm since the evolution of population. In order to speed up the search and obtain better feasible solution, this thesis, aimed at personal best, puts forward elite crossover algorithm for searching better feasible solution.

**E.g.2:** As showed in Figure 2.  $p_a$ ,  $p_b$  are two feasible solutions in a personal best set, among which,  $\Delta$  is a random number and  $p'$  a feasible solution from elite crossover algorithm.

Elite crossover algorithm chooses two better individuals in a population to conduct crossover operation and generate new individuals; meanwhile, reserves the same position information of excellent individuals. Continuous identical segments of excellent individuals can be expressed as good continuity of some workpieces in processing order during scheduling sort, the destruction of which may reduce quality of feasible solution. Therefore, elite crossover algorithm maintains the identical segments information of individuals, and copies the information to new individuals.

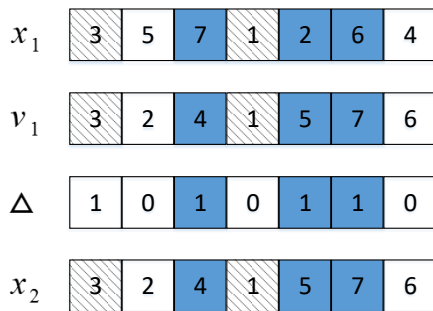


Figure 1. Updating Operation

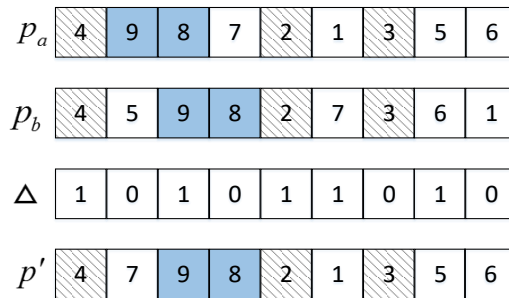


Figure 2. Elite Crossover Operation

### 3.6 Double Insertion Disturbance Algorithms

In elite crossover algorithm, global best serves as the optimal global solution, but if all particles approach to global best with certain velocity, particles may easily fall into local optimum. Therefore, the thesis introduces double insertion disturbance algorithm to prevent the emergence of local optimum. If global best does not update in many times of iterative processes, it indicates that the particle may be trapped in local optimum when the application of double insertion disturbance algorithm can change this stagnant status.

**E.g.3:**  $x$  is the current global best and  $k_1 = 2$ ,  $k_2 = 6$ ,  $k_3 = 7$  and  $k_4 = 10$  are the insertion positions. As Figure 3 displays, there are four exchange intervals  $[1,2]$ ,  $[2,6]$ ,  $[6,7]$  and  $[7,10]$ . If the first interval exchanges with the fourth interval, a new feasible solution  $x'$  can be obtained.

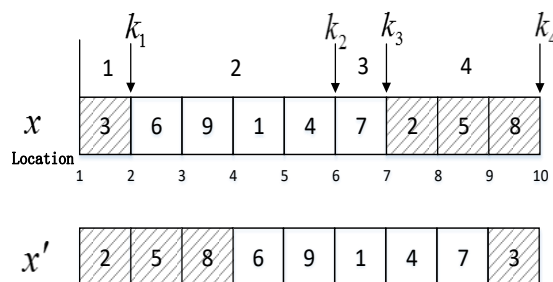


Figure 3. Double Insertion Perturbation Algorithm

The traditional insertion algorithm is to randomly select two insertion places, and insert back-position information to forth-position, with a total of  $(n-2)(n-1)/2$  insertion methods. While double insertion disturbance algorithm randomly divides particles into 3, 4 or 5 intervals and randomly exchanges information of any two intervals. On the basis of double insertion disturbance algorithm, if it is divided into 5 intervals, it totally has  $5 \times (n-2)(n-1)(n-3)(n-4)/12$  exchange methods, which has outnumbered the traditional insertion algorithm, thereby the algorithm can help particles jump out of local optimum and expand search range of particles.

#### 4. Algorithm Performance Indices

There is not a unified test case set in flexible flow shop problem, thus in this experiment, the process time matrix of workpiece  $PT$  is randomly created, and the process time interval is  $[1, 20]$ . The Due date of workpiece is randomly created in the interval  $[P(1-E-R/2), P(1-E+R/2)]$ , among which,  $P$  stands for final completion time of all workpieces, i.e. *makespan*;  $E$  and  $R$  are two random numbers in the interval  $[0,1]$ , which is the same as reference [21], and  $E=0.3$ ,  $R=0.9$  in the thesis.

##### 4.1 Algorithm Performance Indices

The thesis compares four performance indices of algorithm: average distance, number of non-dominated solution, proportion of non-dominated solution and average relative deviation.

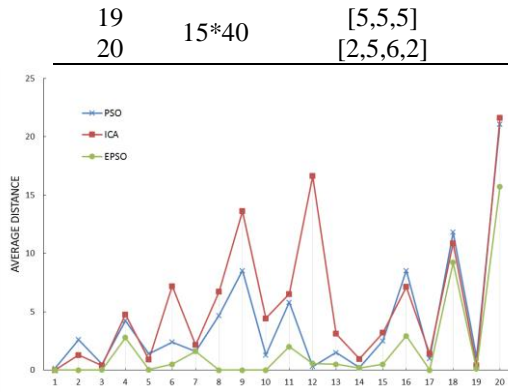
##### 4.2 Algorithm Comparison

To prove accuracy and validity of EPSO algorithm in this thesis, in the PC whose CPU is Intel(R) Core(TM) i5-3470 3.20GHz and memory is over 8G, Matlab 2014a is applied as the experiment environment to achieving EPSO algorithm and comparing PSO algorithm and ICA algorithm. Parameters of EPSO algorithm are decided by experience. Probability of double insertion disturbance algorithm is  $p_{di}=0.01$ , and algorithm iteration number is  $TotalIt=100$ .

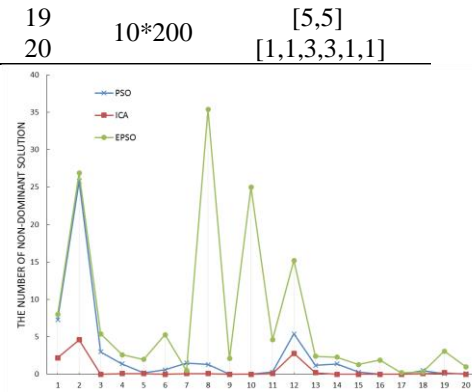
**Table 2. Problem Scale**

Small scale problem			Large scale problem		
Group	scale	parallel machine	Group	scale	parallel machine
1		[2,3]	1		[2,1,2]
2	5*5	[4,1]	2	5*50	[4,1]
3		[3,2]	3		[2,3]
4	5*10	[1,2,2]	4	5*80	[4,1]
5		[2,3]	5	5*100	[2,2,1]
6	5*20	[1,4]	6	5*100	[1,3,1]
7		[3,2]	7		[2,3]
8	5*30	[4,1]	8	5*150	[1,3,1]
9		[1,1,3]	9	5*200	[2,3]
10	5*40	[4,1]	10	5*200	[2,2,1]
11		[2,3,1,4]	11		[5,5]
12	10*10	[5,5]	12	10*50	[3,2,3,2]
13		[2,3,5]	13		[2,2,2,2,2]
14	10*20	[4,6]	14	10*80	[3,3,3,1]
15		[2,5,3]	15		[5,5]
16	10*30	[2,3,3,2]	16	10*100	[2,3,3,2]
17		[5,5]	17		[4,2,4]
18	10*40	[2,3,3,2]	18	10*150	[3,2,2,3]





**Figure 4. Average Distance in Small Problem Scale**



**Figure 5. The Number of Non-Dominated Solutions in Small Problem Scale**

There are 20 groups of comparative experiment data aimed at small-scale and large-scale flexible flow shop problems. As showed as Table2, problem scale (a\*b) is expressed as machine number\*workpiece number and parallel machine number stands for the number of parallel machines in each processing stage. For instance, if problem scale is 5\*10, and parallel machine number is [3,2], it represents that this problem includes 5 processing machines and 10 workpieces to be processed; besides, each workpiece needs 2 processing steps, the first stage including 3 parallel machines and the second stage 2 parallel machines.

There are 40 groups of comparative experiment, each of which involves experiment results of 20-time independent operations. For the model of small-scale problem, by respectively calculating average distance of the solution set of PSO algorithm, ICA algorithm and EPS algorithm, the comparative figure of experiment results is illustrated as Figure 4. Through respectively calculating the number of non-dominated solution in the solution set of PSO algorithm, ICA algorithm and EPS algorithm, the comparative figure of experiment results is displayed as Figure 5.

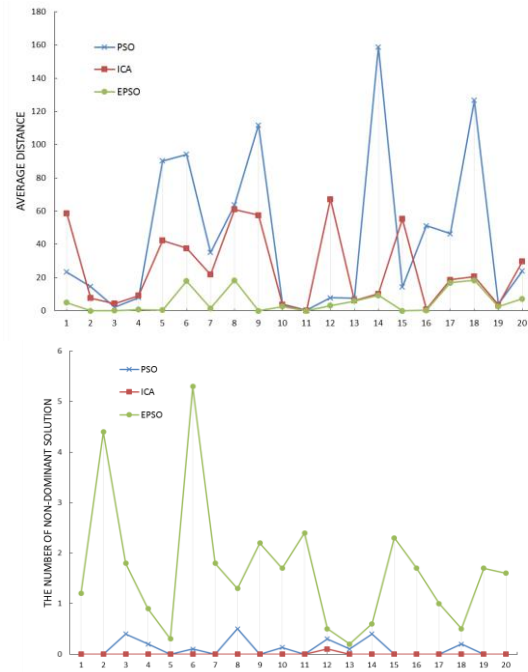
**Table 3. The Proportion of Non-dominated Solutions in Small Problem Scale**

Group	Algorithm		PSO			ICA			EPSO		
	Scale	Machine	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX	AVG
1	5*5	[2,3]	50%	89%	87%	0%	43%	28%	50%	100%	95%
2		[4,1]	77%	100%	95%	12%	30%	17%	96%	100%	99%
3		[3,2]	0%	100%	33%	0%	0%	0%	4%	100%	39%
4	5*10	[1,2,2]	0%	100%	38%	0%	17%	2%	16%	100%	59%
5		[2,3]	0%	67%	7%	0%	17%	2%	12%	100%	25%
6	5*20	[1,4]	0%	75%	8%	0%	0%	0%	0%	100%	43%
7		[3,2]	0%	45%	12%	0%	25%	3%	57%	90%	74%
8		[4,1]	0%	26%	3%	0%	1%	0%	38%	100%	69%
9	5*30	[1,1,3]	0%	0%	0%	0%	0%	0%	17%	45%	38%
10		[4,1]	0%	0%	0%	0%	0%	0%	66%	93%	77%
11	10*10	[2,3,1,4]	0%	50%	7%	0%	20%	2%	0%	100%	30%
12		[5,5]	3%	100%	58%	0%	5%	2%	70%	100%	95%
13	10*20	[2,3,5]	0%	100%	17%	0%	12%	1%	0%	100%	25%
14		[4,6]	0%	100%	24%	0%	0%	0%	0%	85%	17%
15	10*30	[2,5,3]	0%	13%	3%	0%	0%	0%	0%	67%	11%
16		[2,3,3,2]	0%	0%	0%	0%	0%	0%	0%	86%	12%

10*40	[5,5]	0%	0%	0%	0%	0%	0%	0%	50%	5%
	[2,3,3,2]	0%	67%	17%	0%	33%	3%	21%	50%	28%
15*40	[5,5,5]	0%	5%	0%	0%	7%	1%	8%	64%	21%
	[2,5,6,2]	0%	33%	3%	0%	0%	0%	0%	100%	45%
Average value				21%		3%		44%		

The average distance is shorter, the solution set is closer to the non-dominated solution set with better optimization effect. The average distance of each algorithm is showed in Figure4. Based on the table, it is found that in the comparative experiment of 20 groups of small-scale problem, the average distance of PSO algorithm and ICA algorithm is rather volatile, the average distance of ICA algorithm is relatively the largest and most of comparative experiment has maximum. However, the average distance in solution set of EPSO algorithm is almost minimum; especially, the first 10 groups of comparative experiment enjoy the best optimization effect. In the 12<sup>th</sup> group of comparative experiment, the average distance in solution set of EPSO algorithm is similar to that of PSO algorithm, because all the comparative experiment data of algorithm are generated randomly. For some experiment data, PSO algorithm is accessible to better optimization effect, so is EPSO algorithm since it is based on PSO algorithm. According to Figure5, it is known that the number of non-dominated solution brought by ICA algorithm accounts for relatively small proportion, with the least number of non-dominated solution. PSO algorithm in the second experiment group can obtain relatively more number of non-dominated solutions, but in most experiment groups, the number of non-dominated solution is relatively rare and EPSO algorithm almost occupies the maximum of number of non-dominated solution in the comparative groups. Viewed from the figure, the number of non-dominated solution of EPSO algorithm is volatile because of random experiment data and great differences in number of non-dominated solution in various scheduling problems. The number of non-dominated solution can not represent the accurate relation between solution set of algorithm and non-dominated solution set, so the proportion of non-dominated solution of each algorithm in the thesis is showed as Table 3. From Table3, in the 5\*5 problem scale, the proportion of non-dominated solution of each algorithm is relatively large due to relatively small-scale problem and feasible solution range. PSO algorithm, ICA algorithm and EPSO algorithm are on the algorithm basis of population iteration, thus in proper iterations, small-scale problem can get better feasible solution. The proportion of non-dominated solution of EPSO algorithm is larger, which includes 3 times of over 90 percent of non-dominated solutions and 6 times of over 60 percent of non-dominated solutions. While the proportion of non-dominated solution of PSO algorithm and ICA algorithm is smaller. Based on the proportion of non-dominated solution of each algorithm, it is noticeable that the proportion of non-dominated solution of standard PSO algorithm and standard EPSO algorithm is relatively small. The proportion of average non-dominated solution in 20 groups of comparative experiment which calculate PSO algorithm, ICA algorithm and EPSO algorithm is 21%, 3% and 44% respectively; therefore, the proportion of EPSO algorithm is obviously larger than the rest. As for small-scale problem, EPSO algorithm in the thesis shares a good performance and significantly enhances feasible solution quality while comparing with PSO algorithm and ICA algorithm.

The 20 groups of large-scale problem in comparative experiment showed as Table2 calculate the average distance and number of non-dominated solution of these three algorithms, which are illustrated as Figure6 and Figure7.



**Figure 6. Average Distance in Large Problem Scale**

**Figure 7. The Number of Non-Dominated Solutions in Large Problem Scale**

As showed in Figure6, the average distance of solution set of PSO algorithm is relatively large with more noticeable fluctuations; meanwhile, the average distance of ICA algorithm enjoys greater volatility but is smaller than that of PSO algorithm. What's more, in terms of large-scale problem, the average distance of solution set in 20 groups of comparative experiment of EPSO algorithm is smaller and more stable than the other two algorithms. From Figure7, the number of non-dominated solution of PSO algorithm and ICA algorithm in large-scale problem is small. The number of non-dominated solution in most comparative groups approaches to zero while the number of non-dominated solution of EPSO algorithm is obviously large. All of these indicate that as for large-scale problem, PSO algorithm and ICA algorithm have smaller search range of feasible range and less number of non-dominated solution. Since EPSO algorithm adds elite crossover algorithm and double insertion disturbance algorithm, it effectively expands search range of feasible solution and enhance feasible solution quality, thus it enjoying more numbers of non-dominated solution and better optimization effect. The proportion of non-dominated solution of each algorithm calculating from number of non-dominated solution and reference set is showed as Table4.

In Table 4, EPSO algorithm is almost accessible to the maximum of proportion of non-dominated solution in all comparative groups in the experiment. And from Table8, the average proportion of non-dominated solution of PSO algorithm, ICA algorithm and EPSO algorithm is 2%, 1% and 28% respectively, thereby the proportion of EPSO algorithm is much larger than that of the rest.

**Table 4. The Proportion of Non-dominated Solutions in Large Problem Scale**

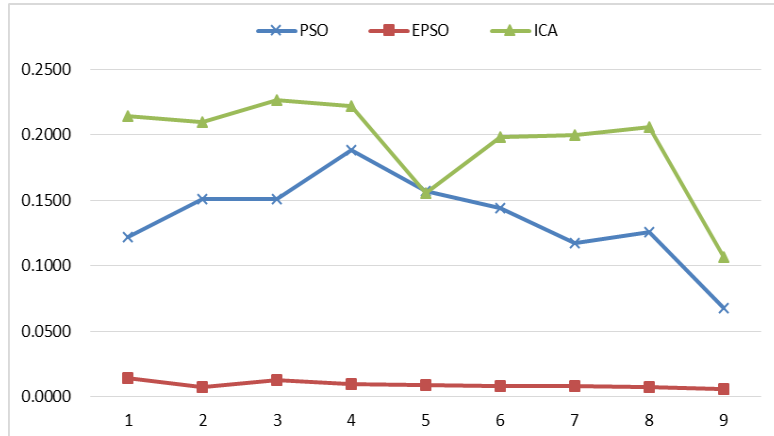
Group	Algorithm		PSO			ICA			IICA		
	Scale	Machine	MIN	MAX	AVG	MIN	MAX	AVG	MIN	MAX	AVG
1	5*50	[2,1,2]	0%	0%	0%	0%	0%	0%	0%	100%	10%
2		[4,1]	0%	0%	0%	0%	0%	0%	0%	86%	39%
3	5*80	[2,3]	0%	0%	0%	0%	0%	0%	0%	100%	45%
4		[4,1]	0%	0%	0%	0%	0%	0%	0%	12%	2%
5	5*100	[2,2,1]	0%	0%	0%	0%	30%	17%	0%	40%	21%
6		[1,3,1]	0%	0%	0%	0%	0%	0%	10%	26%	13%
7	5*150	[2,3]	0%	0%	0%	0%	0%	0%	0%	14%	3%
8		[1,3,1]	0%	0%	0%	0%	0%	0%	10%	100%	98%
9	5*200	[2,3]	0%	0%	0%	0%	0%	0%	67%	100%	97%
10		[2,2,1]	0%	0%	0%	0%	0%	0%	0%	100%	59%
11	10*50	[5,5]	0%	0%	0%	0%	0%	0%	7%	100%	22%
12		[3,2,3,2]	0%	50%	8%	0%	17%	2%	0%	50%	5%
13	10*80	[2,2,2,2,2]	0%	33%	3%	0%	0%	0%	0%	22%	2%
14		[3,3,3,1]	0%	100%	20%	0%	0%	0%	0%	100%	10%
15	10*100	[5,5]	0%	0%	0%	0%	0%	0%	17%	100%	65%
16		[2,3,3,2]	0%	0%	0%	0%	0%	0%	0%	100%	30%
17	10*150	4,2,4]	0%	0%	0%	0%	30%	12%	30%	42%	31%
18		[3,2,2,3]	0%	100%	15%	0%	0%	0%	0%	0%	0%
19	10*200	[5,5]	0%	0%	0%	0%	0%	0%	0%	100%	59%
20		[1,1,3,3,1,1]	0%	0%	0%	0%	0%	0%	23%	100%	53%
Average value					2%	1%			28%		

According to different weight coefficient combinations, aimed at 10 groups of flexible slow shop problem cases with diversified scales, objective function values of PSO algorithm, ICA algorithm and EPSO algorithm are calculated respectively, and the calculating results are illustrated as Table 5.

There are 9\*10groups of comparative data in Table 5, among which, EPSO algorithm gets 84 times of the minimum of average relative deviation; for average relative deviation of algorithm in each weight group, EPSO algorithm obtains the minimum in 9 groups and achieves the best optimization effect of algorithm. The average relative deviation table made from Table 5 is showed as Figure 8.

**Table 5. The Calculation Results of Each Algorithm under Combination Weights**

Group	Weight	Scale	5*10	5*30	5*50	5*100	5*150	10*10	10*50	10*100	10*150	10*200	Average
1	$\omega_1 = 0.1$	PSO	0.0692	0.0210	0.5157	0.1588	0.0787	0.1814	0.0787	0.0262	<b>0.0118</b>	0.0797	0.1221
	$\omega_2 = 0.9$	ICA	0.0880	0.0432	0.3480	0.7756	0.1000	0.5401	0.1064	0.0453	0.0140	0.0842	0.2145
		IPSO	<b>0.0445</b>	<b>0.0257</b>	<b>0.0226</b>	<b>0.0098</b>	<b>0.0000</b>	<b>0.0185</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0000</b>	0.0161	<b>0.0035</b>
2	$\omega_1 = 0.2$	PSO	<b>0.0161</b>	0.0315	0.1095	0.9161	0.0766	0.1442	0.0830	0.0226	0.0165	0.0854	0.1502
	$\omega_2 = 0.8$	ICA	0.0762	0.0391	0.5002	0.1570	0.1106	0.2860	0.1170	0.0336	0.0174	0.1045	0.1442
		IPSO	0.0480	<b>0.0146</b>	<b>0.0169</b>	<b>0.0133</b>	<b>0.0000</b>	<b>0.0030</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0005</b>	<b>0.0121</b>	<b>0.0000</b>
3	$\omega_1 = 0.3$	PSO	0.0391	0.1326	0.7893	0.0314	0.0787	0.2195	0.0872	0.0238	0.0201	0.0889	0.1511
	$\omega_2 = 0.7$	ICA	0.0658	0.0466	1.2956	0.1911	0.1021	0.3043	0.1043	0.0490	0.0252	0.0834	0.2267
		IPSO	<b>0.0156</b>	<b>0.0222</b>	<b>0.0239</b>	<b>0.0130</b>	<b>0.0000</b>	<b>0.0319</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0007</b>	<b>0.0168</b>	<b>0.0000</b>
4	$\omega_1 = 0.4$	PSO	0.0906	0.0246	0.4932	0.9014	0.0702	0.0939	0.0745	0.0346	0.0245	0.0744	0.1882
	$\omega_2 = 0.6$	ICA	0.0535	0.0291	0.3543	1.1636	0.1128	0.1568	0.1064	0.1307	0.0274	0.0834	0.2218
		IPSO	<b>0.0189</b>	<b>0.0235</b>	<b>0.0192</b>	<b>0.0106</b>	<b>0.0000</b>	<b>0.0046</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0030</b>	<b>0.0159</b>	<b>0.0000</b>
5	$\omega_1 = 0.5$	PSO	0.0938	0.0357	0.3318	0.5959	0.0809	0.0392	0.0809	0.0214	0.1999	0.0901	0.1569
	$\omega_2 = 0.5$	ICA	0.0469	0.0425	0.6384	0.0589	0.1085	0.1549	0.0936	0.0440	0.2773	0.0911	0.1556
		IPSO	<b>0.0092</b>	<b>0.0100</b>	<b>0.0214</b>	<b>0.0054</b>	<b>0.0000</b>	<b>0.0260</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0030</b>	<b>0.0100</b>	<b>0.0015</b>
6	$\omega_1 = 0.6$	PSO	0.0095	0.0278	0.2119	0.8317	0.0766	0.0383	0.0830	0.0421	0.0362	0.0854	0.1443
	$\omega_2 = 0.4$	ICA	0.0530	0.0433	0.4708	0.5681	0.1000	0.0977	0.1000	0.0461	0.4196	0.0869	0.1985
		IPSO	<b>0.0158</b>	<b>0.0154</b>	<b>0.0194</b>	<b>0.0112</b>	<b>0.0000</b>	<b>0.0071</b>	<b>0.0000</b>	<b>0.0030</b>	<b>0.0079</b>	<b>0.0000</b>	<b>0.0080</b>
7	$\omega_1 = 0.7$	PSO	0.2845	0.0207	0.1099	0.1953	0.0787	0.0232	0.0809	0.0292	0.2701	0.0819	0.1174
	$\omega_2 = 0.3$	ICA	0.5287	0.3460	0.2987	0.4276	0.0894	0.0601	0.0851	0.0451	0.0279	0.0905	0.1999
		IPSO	<b>0.0159</b>	<b>0.0051</b>	<b>0.0214</b>	<b>0.0127</b>	<b>0.0000</b>	<b>0.0078</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0012</b>	<b>0.0188</b>	<b>0.0000</b>
8	$\omega_1 = 0.8$	PSO	0.1905	0.2229	0.0780	0.2630	0.1000	<b>0.0023</b>	0.0894	0.1241	0.1187	0.0673	0.1256
	$\omega_2 = 0.2$	ICA	0.2786	0.1982	0.1709	0.9422	0.1085	0.0194	0.1000	0.0297	0.1312	0.0829	0.2062
		IPSO	<b>0.0126</b>	<b>0.0103</b>	<b>0.0218</b>	<b>0.0113</b>	<b>0.0000</b>	0.0150	<b>0.0000</b>	<b>0.0000</b>	<b>0.0002</b>	<b>0.0007</b>	<b>0.0000</b>
9	$\omega_1 = 0.9$	PSO	0.0224	<b>0.0158</b>	0.0441	0.2643	0.0830	0.0057	0.0830	0.0388	0.0322	0.0839	0.0673
	$\omega_2 = 0.1$	ICA	0.0406	0.0539	0.0828	0.4865	0.1085	0.0198	0.1128	0.0423	0.0402	0.0749	0.1062
		IPSO	<b>0.0125</b>	0.0209	<b>0.0132</b>	<b>0.0071</b>	<b>0.0000</b>	<b>0.0050</b>	<b>0.0000</b>	<b>0.0000</b>	<b>0.0001</b>	<b>0.0022</b>	<b>0.0000</b>



**Figure 8. The Average Relative Deviation Comparison Figure**

Based on Figure8, from the calculating results of total average relative deviation considering 9 groups of weight coefficients, it is accessible to the performance rank of each algorithm:  $ICA < PSO < EPSO$ . It clarifies that PSO algorithm is better than ICA algorithm while ICA algorithm enjoying the worst performance, and both of the algorithms share obvious fluctuations in average relative deviation. EPSO algorithm boasts the best performance with smaller and inconspicuous fluctuations in average relative deviation, which can be suitable to various scheduling problems with different weight coefficients. The better performance of EPSO algorithm attributes to the addition of elite crossover algorithm and double insertion perpetuation algorithm, so as to dramatically expand search range of algorithm and facilitate feasible solution quality.

## 5. Conclusion

The thesis proposes elite particle swarm optimization to solve bi-criteria no-wait flexible flow shop problem, whose optimization objective is to minimize the maximum completion time and the maximum delay time. To enhance quality of initial population and maintain random characteristic, at the initial stage of algorithm, it is based on five initial feasible solutions of five modified heuristic algorithms, and the rest initial populations are randomly produced among feasible solution ranges. For the purpose of expanding search range of feasible solution, the thesis raises elite crossover algorithm aimed at particle individual, and reserves fragments of the continuously identical workpieces among excellent individuals in updating operation, thus avoiding the destruction of good continuity among workpieces. In addition, the thesis introduces double insertion disturbance algorithm aimed at global best in order to prevent population from a local optimum, expand search range of feasible solution and optimize feasible solution quality. For the sake of evaluating performance of EPSO algorithm, PSO algorithm and ICA algorithm, by setting the problem model into small-scale problem and large-scale problem in the simulation experiment, 20 groups of problem case data are generated; the aforesaid algorithms are respectively applied to having solutions of each problem and compares average distance, number of non-dominated solution, proportion of non-dominated solution and average relative deviation. The experiment results prove that EPSO algorithm in the thesis is superior to other algorithms in terms of each performance index and, its feasible solution set is closer to non-dominated solution set. In future research, non-dominated solution or addition of other constraints, such as workpiece release time, processing priority limit and so on, will be considered and studied.

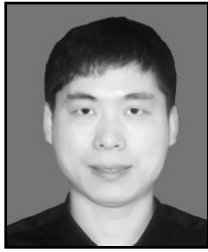
## Acknowledgements

The research work was supported by National Natural Science Foundation of China under Grant No.61262006 and No.61540050, Major Applied Basic Research Program of Guizhou Province under Grant No.JZ20142001 and the Science and Technology Foundation of Guizhou Province under Grant No.LH20147636.

## References

- [1] K. Wang and S. H. Choi, "A holonic approach to flexible flow shop scheduling under stochastic processing times", *Computers & Operations Research*, vol. 43, (2014), pp. 157-168.
- [2] D. Tang, M. Dai and M. A. Salido, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization", *Computers in Industry*, (2015).
- [3] T. S. Arthanari and K. G. Ramamurthy, "An extension of two machines sequencing problem", *Opsearch*, vol. 8, no. 1, (1971), pp. 10-22.
- [4] M. S. Salvador, "A solution to a special class of flow shop scheduling problems", *Symposium on the theory of scheduling and its applications*, Springer Berlin Heidelberg, (1973).
- [5] A. Guinet, M. M. Solomon and P. K. Kedia, "A computational study of heuristics for two-stage flexible flowshops", *International Journal of Production Research*, vol. 34, no. 5, (1996), pp. 1399-1415.
- [6] G. B. Mainieri and D. P. Ronconi, "New heuristics for total tardiness minimization in a flexible flow-shop", *Optimization Letters*, vol. 7, no. 4, (2013), pp. 665-684.
- [7] J. Jungwattanakit, M. Reodecha and P. Chaovalitwongse, "Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria", *The International Journal of Advanced Manufacturing Technology*, vol. 37, no. 3-4, (2008), pp. 354-370.
- [8] M. Zandieh, S. M. T. F. Ghomi and S. M. M. Hussein, "An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times", *Applied Mathematics and Computation*, vol. 180, no. 1, (2006), pp. 111-127.
- [9] M. S. Nagano, A. A. Da Silva, L. A. N. Lorena, "An evolutionary clustering search for the no-wait flow shop problem with sequence dependent setup times", *Expert Systems with Applications*, vol. 41, no. 8, (2014), pp. 3628-3633.
- [10] S. U. Sapkal and D. Laha, "A heuristic for no-wait flow shop scheduling", *The International Journal of Advanced Manufacturing Technology*, vol. 68, no. 5-8, (2013), pp. 1327-1338.
- [11] Q. Pan, W. Wang and J. Zhu, "Some meta-heuristics for no-wait flow shop problem", *Computer integrated manufacturing system-Beijing*, vol. 13, no. 5, (2007), p. 967.
- [12] H. Röck, "The three-machine no-wait flow shop is NP-complete", *Journal of the ACM (JACM)*, vol. 31, no. 2, (1984), pp. 336-345.
- [13] M. Khalili and B. Naderi, "A bi-objective imperialist competitive algorithm for no-wait flexible flow lines with sequence dependent setup times", *The International Journal of Advanced Manufacturing Technology*, vol. 76, no. 1-4, (2015), pp. 461-469.
- [14] Z. Liu, J. Xie and J. Li, "A heuristic for two-stage no-wait hybrid flow shop scheduling with a single machine in either stage", *Tsinghua Science and Technology*, vol. 8, no. 1, (2003), pp. 43-48.
- [15] N. Moradinab, R. Shafaei and M. Rabiee, "No-wait two stage hybrid flow shop scheduling with genetic and adaptive imperialist competitive algorithms", *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 25, no. 2, (2013), pp. 207-225.
- [16] M. Rabiee, R. S. Rad and M. Mazinani, "An intelligent hybrid meta-heuristic for solving a case of no-wait two-stage flexible flow shop scheduling problem with unrelated parallel machines", *The International Journal of Advanced Manufacturing Technology*, vol. 71, no. 5-8, (2014), pp. 1229-1245.
- [17] W. H. M. Raaymakers and J. A. Hoogeveen, "Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing", *European Journal of Operational Research*, vol. 126, no. 1, (2000), pp. 131-151.
- [18] F. Jolai, S. Sheikh and M. Rabbani, "A genetic algorithm for solving no-wait flexible flow lines with due window and job rejection", *The International Journal of Advanced Manufacturing Technology*, vol. 42, no. 5-6, (2009), pp. 523-532.
- [19] R. H. Huang, C. L. Yang and Y. C. Huang, "No-wait two-stage multiprocessor flow shop scheduling with unit setup", *The International Journal of Advanced Manufacturing Technology*, vol. 44, no. 9-10, (2009), pp. 921-927.
- [20] D. Danneberg, T. Tautenhahn and F. Werner, "A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size", *Mathematical and Computer Modelling*, vol. 29, no. 9, (1999), pp. 101-126.
- [21] A. Allahverdi and T. Aldowaisan, "No-wait flow shops with bicriteria of makespan and maximum lateness", *European Journal of Operational Research*, vol. 152, no. 1, (2004), pp. 132-147.
- [22] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", *Proceedings of the sixth international symposium on micro machine and human science*, (1995).

## Authors



**Yongbin Qin**, He received his B.S. degree in Computer Science from the Jinan University, China, in 2003 and the Mphil. and PhD. degrees in the Computer Science and Technology from Guizhou University, China, in 2007 and 2011. In 2007, he joined Guizhou University, China, as a lecturer. Since the year 2011, he has been an associate professor in Guizhou University. He is an active researcher on the area of Intelligent Computing, Machine Learning, and Algorithm Design.



**Haiyue Zhang**, She received her B.S. degree in Computer Science from the Nanjing Forestry University, China, in 2013 and the Mphil degree in the Computer Science and Technology from Guizhou University, China, in 2016. She is an excellent student and doing a good research on the area of Algorithm Design.