

## A Fuzzy Immune GA Scheduling Framework for JSSP

Deyin Ma<sup>1</sup>, Xuming Han<sup>2</sup>, Yanchun Liang<sup>3</sup>, and Xiaohu Shi<sup>4\*</sup>

<sup>1,2</sup>*School of Computer Science and Engineering, Changchun University of Technology*

<sup>3,4</sup>*College of Computer Science and Technology, Jilin University, Changchun, China*

<sup>1</sup>*madeyinzy521@gmail.com*, <sup>2</sup>*hanxuming@163.com*, <sup>3</sup>*ycliang@jlu.edu.cn*,  
<sup>4</sup>*shixh@jlu.edu.cn*

### Abstract

*Focused on partly flexible job shop scheduling problems, a framework is proposed based on fuzzy immune GA scheduling method. Firstly, we develop a dispatching rule method to evaluate processing devices based on fuzzy weight according to multiple objectives. Based on the evaluation results, the devices are selected to reduce the redundant ones. Then a fuzzy immune genetic algorithm is proposed to deal with the scheduling problems, in which the fuzzified maximum average satisfaction is used to evaluate the solution. To test our proposed method, it is applied to two real job scheduling problems. Experiment results show the effectiveness of our method.*

**Keywords:** *JSSP, fuzzy weight, immune genetic algorithm, maximum satisfaction, dispatching rule*

### 1. Introduction

In conventional scheduling problems, the processing parameters are always considered to be exact values. There are many algorithms to solve the scheduling problems, *e.g.*, GA [1], particle swarm [2], ANN [3] *etc.* However, in real cases, the exact processing time and production delivery date are often difficult to obtain because there might exist some uncertain parameters, such as the potential of machine faults, the differentiation of workers' proficiency, the variation of environmental situations, and so on. Therefore, the manager of workshop can only get approximate values or possible variation intervals and based on these to make decisions. To deal with such uncertain cases, convention methods use exact values to represent those uncertain values approximately. While there are many shortcomings in this way, one of the most important issues is that the solution will change with the model's variation. Another kind of methods is to use random probability distribution function representing the uncertain parameters. This kind of method needs the historical data of parameters which sometimes are difficult to obtain [4].

As the development of fuzzy theory, it has been widely used to solve the uncertain problems. For example, the production delivery date could be fuzzified into different degrees of the most optimistic date, the most possible date, and the most pessimistic date. It is more like to agree with the practice by dealing with uncertain parameters according to fuzzy theory. This kind of scheduling problem is also called fuzzy scheduling problem and has become an important branch of uncertain scheduling problems [5]. Fuzzy scheduling problems can be solved by many algorithms, *e.g.* fuzzy GA [6], fuzzy particle swarm [7] *etc.* But in practical

---

\* Corresponding Author

scheduling, big number of optional flexible devices, variety in workpiece and complicated process all can lead to long scheduling time and low satisfaction rate [8]. Therefore the key of scheduling is to solve the foregoing problems. Focused on partly flexible job shop scheduling problems, in this paper, we develop a comprehensive fuzzy weight based dispatching rule method to evaluate the fuzzy weights of the devices and process routes according to multiple objectives, namely that the maximum makespan, the average flowing time, the total delaying time and the total load of machines. Based on the evaluation results, the devices and process routes are initially selected to reduce the redundant ones and improve the scheduling efficiency. Then we propose an immune genetic algorithm for the detail scheduling by integrating the membership functions of fuzzified makespan and the expected production delivery date as the objective function. To test our developed framework, it has been applied to two real experiments. Experimental results show that our method is effective.

## 2. Fuzzy Weight Based Device Optimization Method

There are many departments in an enterprise, and each of them has its own interests. Therefore they have different expectations for the shop scheduling of the enterprise. For instance, the sales department wants the production could be delivered on time; while the manufacturing department cares more about reducing of the cost; and the production department hopes the maximum of production efficiency. Some commonly used objective functions are Makespan, Average flowing time, Total extension time, Total load of machine, The choke point of machine load, Production cost, and so on [9]. Then the key point of the shop scheduling method is to balance the benefits of different departments [10]. Hence in this paper, a comprehensive fuzzy weight based dispatching rule method is developed to balance different objectives by integrating them with different weights, and reduce those redundant devices and process routes accordingly.

In real job scheduling problems, it is hard to give an exact evaluation criterion for most of the optimized objectives [11]. Therefore a comprehensive fuzzy weight based dispatching rule method is proposed to evaluate the machines according to multi-objectives and then reduce those redundant devices for optimization. The main idea is to fuzzify different objectives into several discrete values associating different weights, and then defuzzify them by evaluating their weights [12]. The details of the method are described as follows:

1. Definite the set of devices (machines) to be evaluated  $M = \{M_1, \dots, M_m\}$ , set an evaluation threshold value  $\lambda$  for  $M$ .
2. Build the set of factors (objectives for optimization)  $F = \{f_1, \dots, f_n\}$ . For example, the elements of the set could be any objective functions mentioned before.
3. Construct the evaluation level set  $V = \{v_1, \dots, v_p\}$ , which consists of the elements of fuzzified discrete values, for example, such as “best”, “better”, “worse”, “worst”, and so on.
4. Build the weight set  $W = \{w_{ij}\}_{m \times n}$ . Here  $w_{ij}$  represents the importance of  $f_j$  for  $M_i$ . For normalization, it should be satisfied that

$$\sum_{j=1}^n w_{ij} = 1 \quad i = 1, 2, \dots, m \quad (1)$$

5. Construct the comprehensive evaluation array  $R = \{r_{ijk}\}_{m \times n \times p}$ , which represents the fuzzy relationship of  $M$  between  $F$  and  $V$ , namely that  $r_{ijk}$  measures how extend that  $f_j$  belongs to  $v_k$  of  $M_i$ . For normalization, the elements of  $R$  should satisfy that

$$\sum_{j=1}^n r_{ijk} = 1 \quad i = 1, 2, \dots, m; \quad k = 1, 2, \dots, p \quad (2)$$

6. Compute the comprehensive fuzzy weight vector  $B = \{ b_{ij} \} m \times p$  according to Eq. (3)

$$b_{ij} = \max_{k \in \{1, 2, \dots, n\}} (\min \{ w_{ij}, r_{ikj} \}) \quad (3)$$

7. Calculate the final evaluation result vector  $E = \{ e_1, \dots, e_m \}$ . The element  $e_i$  represents the wellness of  $M_i$  and is computed according to Eq. (4)

$$e_i = \frac{\sum_{j=1}^p v_j b_{ij}}{\sum_{j=1}^p b_{ij}} \quad i = 1, 2, \dots, m \quad (4)$$

8. Make decision according to vector  $E$ : if  $e_i > \lambda$  then retain device  $M_i$ , otherwise delete it.

### 3. The Fuzzy Immune GA Scheduling Algorithm

#### 3.1. The Fuzzified Objective Function for Job Scheduling Problem

The uncertain characteristics on processing parameters are quite evident in practical workshop scheduling problem. With the reasons of processing machinery, environment or the human factor, in general, many processing parameters are unable or hard to obtain. For example, the processing time and the delivery time are difficult to get the accurate values. However, those factors are always varied within an interval. Influenced by the factors derived from machining process or people's expectation, generally we could get the most optimistic time, the most pessimistic time, the most promising time, the most satisfied time, and the acceptable time interval of them. Therefore, to deal with the uncertainty of the problems, the processing parameters are described according to fuzzy theory [13].

Firstly, the processing time is fuzzified using a triangular membership function, which could be represented by a fuzzy triple  $(p^1, p^2, p^3)$ . The three elements of the triples mean the most optimistic time, the most satisfied time, and the most pessimistic time, respectively. Take the processing time of the  $j$ th working procedure of workpiece  $J_i$  processed on the machine  $M_k$ , namely  $p_{ijk}$  as an example, the triangular membership function is depicted by Figure 1 [14].

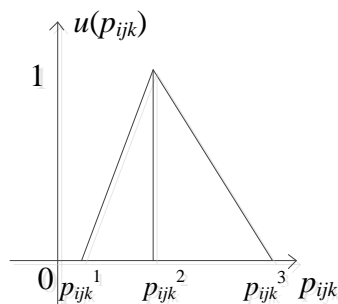


Figure 1. Triangular Membership Function

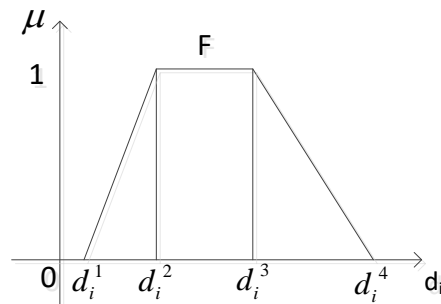


Figure 2. Trapezoidal Membership Function

And then, the delivery time is fuzzified by a trapezoidal membership function. Similarly, it could be represented by a fuzzy four-tuple  $(d_1, d_2, d_3, d_4)$  and the four elements of which mean the most optimistic time, the upper and lower bound of the acceptable time interval, and the most pessimistic time, respectively. Figure 2 shows the trapezoidal membership function of  $D_i$ : the delivery time of the  $i$ th

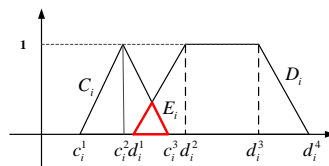
workpiece Ji [15]. Table. 1 shows the fuzzification result of processing time, where “-“ means that the operation cannot be processed on the corresponding machine .

**Table 1. Fuzzification of Processing Time**

Job	operation	Optional machine fuzzy processing time				Fuzzy delivery date of jobs
		$M_1$	$M_2$	...	$M_m$	
$J_1$	$O_{1,1}$	$p_{1,1,1}^1, p_{1,1,1}^2, p_{1,1,1}^3$	—	...	$p_{1,1,m}^1, p_{1,1,m}^2, p_{1,1,m}^3$	$d_1^1, d_1^2, d_1^3, d_1^4$
	...	...	...	...	...	
	$O_{1,n1}$	—	$p_{1,n1,2}^1, p_{1,n1,2}^2, p_{1,n1,2}^3$	...	$p_{1,n1,m}^1, p_{1,n1,m}^2, p_{1,n1,m}^3$	
$J_2$	$O_{2,1}$	$p_{2,1,1}^1, p_{2,1,1}^2, p_{2,1,1}^3$	$p_{2,1,2}^1, p_{2,1,2}^2, p_{2,1,2}^3$	...	—	$d_2^1, d_2^2, d_2^3, d_2^4$
	...	...	...	...	...	
	$O_{2,n2}$	$p_{2,n2,1}^1, p_{2,n2,1}^2, p_{2,n2,1}^3$	$p_{2,n2,2}^1, p_{2,n2,2}^2, p_{2,n2,2}^3$	...	—	
...	...	...	...	...	...	...
$J_n$	$O_{n,1}$	$p_{n,1,1}^1, p_{n,1,1}^2, p_{n,1,1}^3$	—	...	$p_{n,1,m}^1, p_{n,1,m}^2, p_{n,1,m}^3$	$d_n^1, d_n^2, d_n^3, d_n^4$
	...	...	...	...	...	
	$O_{n,nn}$	—	$p_{n,nn,2}^1, p_{n,nn,2}^2, p_{n,nn,2}^3$	...	$p_{n,nn,m}^1, p_{n,nn,m}^2, p_{n,nn,m}^3$	

In this paper, the processing time and delivery time are both considered for the optimization. Therefore, a called “average satisfied degree” is proposed to integrate the above two indexes [16]. Suppose that the jth operation of job Ji is processed on machine ij then the sum of the fuzzy processing time of all the operations of this job

is  $(c_i^1, c_i^2, c_i^3)$ , where  $c_i^k = \sum_{j=1}^{n_i} P_{i,j,i_j}^k$ ,  $k=1, 2, 3$ . Denote SCi as the area modeled by the triangular membership function  $(c_i^1, c_i^2, c_i^3)$ , SDi the area modeled by the trapezoidal membership function  $(d_i^1, d_i^2, d_i^3, d_i^4)$ , and SEi the intersected area of SCi and SDi, respectively (Figure 3). Then the satisfaction ratio of this job is defined as  $AI_i = SE_i / SC_i$  [17].



**Figure 3. The Diagram of SCi, SDi and SEi**

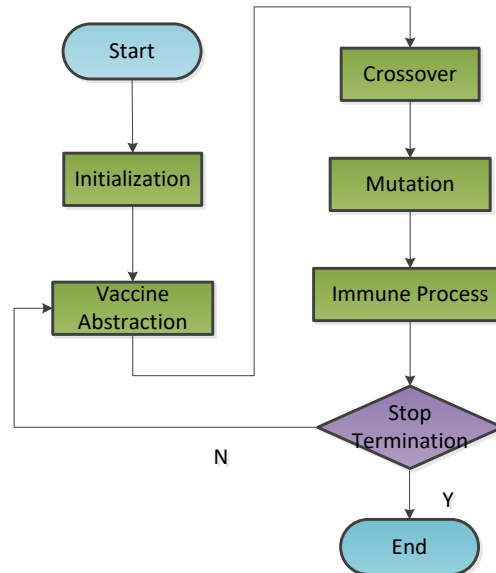
Then the satisfaction ratio of the overall scheduling is

$$AI = \frac{1}{n} \cdot \sum_{i=1}^n AI_i = \frac{1}{n} \cdot \sum_{i=1}^n \frac{S_{E_i}}{S_{C_i}} \quad (5)$$

### 3.2. Immune GA

Traditional GA has problems of prematurity, slow convergence and instable qualities of solutions [18]. This paper combines the memory function of immune algorithm with genetic algorithm (GA), in order to improve the qualities of solutions,

and speed up the convergence. The main idea of the algorithm is to perform repetitive iterations such as vaccine abstraction, selection, crossover, mutation, and immune process on each generation. When the times of iteration reach the predefined number, the chromosome with the best fitness as the best scheduling [19]. The algorithm is illustrated in Figure4 and the details are described as follows:

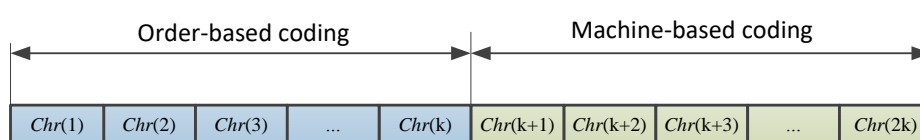


**Figure 4. The Structure of Immune GA**

(1) Coding schedule

Flexible Job Scheduling Problems (FJSPs) require not only the determination of the processing sequence of all operations, but also the dispatch of suitable machine of each operation. Therefore, only using order-based coding is not enough. Based on the characters of FJSP, an extended order-based coding is adopted. The coding is composed of two sections, the former part is order-based, used to determine the processing sequence of operations; the latter part is machine-based, used to dispatch suitable machine to each operation [20]. Suppose there are  $n$  jobs to be processed on  $m$  machines, respectively. Denote  $k_i$  as the number of operations of

the  $i$ th job, so the total number of the operations of all jobs should be  $K = \sum_{i=1}^n k_i$ . Then both parts of the coding of chromosome  $Chr$  have  $K$  elements. The  $j$ th element of order-based part  $Chr(j) \in \{1, 2, \dots, K\}$  means that the  $j$ th processed operation is the  $Chr(j)$ th one. And the  $j$ th element of machine-based part  $Chr(K+j) \in \{1, 2, \dots, m\}$  means that the  $j$ th processed operation is operated on the  $Chr(K+j)$ th machine (See Figure 5).



**Figure 5. Coding Designation**

(2) Calculation of fitness

Given a chromosome, we can obtain the operation sequence from the order-based part and obtain the machine dispatching from the machine-based part according to the coding schedule described above. Namely that a chromosome is decoded into a schedule with job sequence and machine dispatching. Then the fitness could be calculated accordingly. Firstly, we can get the fuzzy processing time of every operation in the schedule according to Table 1. Suppose that the  $j$ th operation  $O_{ij}$  of the  $i$ th job  $J_i$  is performed on machine  $i_j$ . From Table 1., we could get the fuzzy processing time of  $O_{ij}$  and the fuzzy due time of  $J_i$ , namely that

$(P_{i,j,i_j}^1, P_{i,j,i_j}^2, P_{i,j,i_j}^3)$  and  $(di1, di2, di3, di4)$ . Then the fuzzy total processing time

of the job  $(ci1,ci2,ci3)$  could be calculated by  $c_i^k = \sum_{j=1}^{n_i} P_{i,j,i_j}^k$   $k=1, 2, 3$ . Therefore the satisfaction rate of the decoded schedule or the fitness of the chromosome AI could be calculated according to Eq. (5).

(3) Vaccine abstracting process

1) Consider the top  $T$  chromosomes with the highest fitness in the pool as a new population, in which there are  $K$  (from 1 to  $K$ ) options for each position of the order-based parts. Then the probability of the  $i$ th gene value being number  $j$  in the population is  $P_{ij}$ :

$$P_{ij} = \frac{1}{T} \sum_{k=1}^T \delta(a_i^k, j) \quad i=1, \dots, K; j=1, \dots, K \quad (6)$$

where  $\delta(a_i^k, j) = \begin{cases} 1 & a_i^k = j \\ 0 & \text{otherwise} \end{cases}$ , and  $a_i^k$  stands for the code of  $i$ th gene of the  $k$ th chromosome.

2) Abstract the genes with  $P_{ij}$  greater than threshold value  $V$  as the vaccine of the population; *i.e.* the final vaccine abstracted is  $IM=(im1, im2, \dots, imK)$ ,

where  $im_i = \begin{cases} j & \max(P_{ij}) \geq V \\ 0 & \text{otherwise} \end{cases}$ , and  $im_i=0$  means that there is no vaccine at the  $i$ th position of the chromosome.

(4) Selection operation

In our method, ranking selection is chosen as the selection operation. After calculating the fitness of all the individuals in the pool, we sort them according to their fitness values. Assume there are  $N$  chromosomes in the population. Denote  $fit_i$  as the fitness of the  $i$ th one. So we have  $fit_1 \geq fit_2 \geq \dots \geq fit_N$ . Then in the selection process, the selected probabilities of individuals are computed according to their ranking number, namely that

$$p_i = \left(1 + \frac{a}{2} - \frac{a \times i}{N+1}\right) \times \frac{1}{N}, \quad i=1,2,\dots,N; 0 \leq a \leq 2 \quad (7)$$

From Eq. (7), it is easy to deduce that  $p_1 \geq p_2 \geq \dots \geq p_N$  and  $\sum_{i=1}^N p_i = 1$ .

(5) Crossover operation

Assume  $Pa_1$  and  $Pa_2$  are the two selected parent chromosomes in selection operation, then in crossover operation, two child chromosomes  $Ch_1$  and  $Ch_2$  will generated from the parent ones. Firstly, two positions  $N_{start}$  and  $N_{end}$  are randomly generated which are set as the start and end points of crossover interval in order-based coding. Denote  $S_1$  and  $S_2$  as the crossover intervals of  $Pa_1$  and  $Pa_2$ , respectively. Then the pseudo-code of crossover process could be described as Figure 6.

```

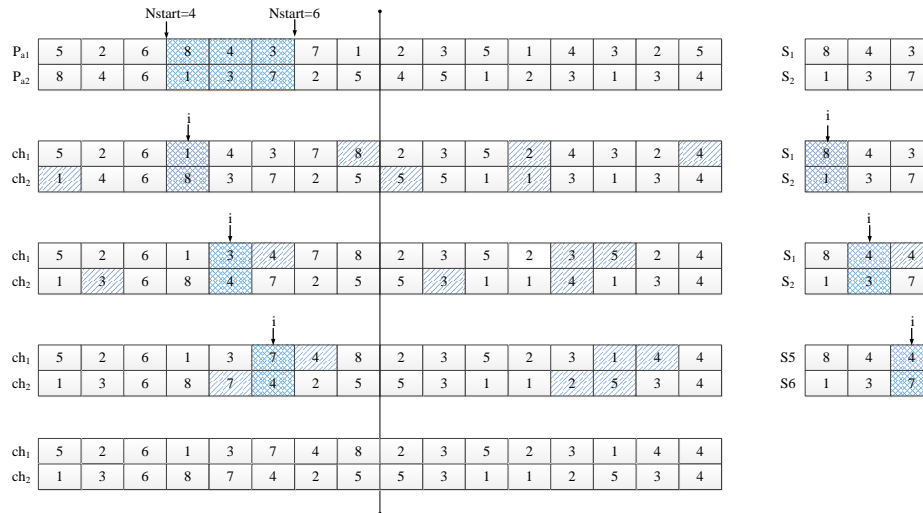
Ch1= Pa1; Ch2= Pa2;
For i=Nstart to Nend
    J1=arg(Pa1(j) =S2 (i- Nstart+1)); J2=arg(Pa2(j) =S1 (i- Nstart+1));
    Swap(Ch1(i), Ch1(J1)); Swap(Ch2(i), Ch2(J2));
    Swap(Ch1(i+K), Ch1(J1+K)); Swap(Ch2(i+K), Ch2(J2+K));
    Swap(Ch1(i+K), Ch2(J2+K)); Swap(Ch2(i+K), Ch1(J1+K));
    If (i ≤J1≤Nend) S1 (J1- Nstart+1)=Ch1(J1)
    If (i ≤J2≤Nend) S2 (J2- Nstart+1)=Ch2(J2)
End For
    
```

**Figure 6. The Pseudo-code of Crossover Process**

Here Swap() is the swapping function. To describe the crossover operation more clearly, an example is given by Figure 7.

(6) Immune operation

The individuals are randomly selected to do the immune operation according to the pre-given probability of immunity. Assume a selected candidate is Ch<sub>im</sub>, which should be vaccinated by the IM=(im1, im2, ..., imK). Denote the newly chromosome is Ch<sub>new</sub>. Then the pseudo-code of immune operation is shown by Figure 8.



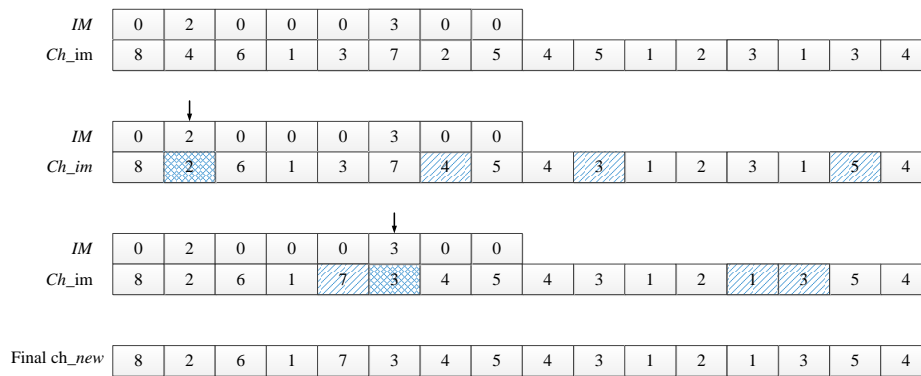
**Figure 7. Crossover Operation**

```

Chnew= Chim
For i=1 to K
    If imi≠0
        J=arg(Chnew(j) = imi);
        Swap(Chnew(i), Chnew(J));
        Swap(Chnew(i+K), Chnew(J+K));
    End if
End for
    
```

**Figure 8. The Pseudo-code of Immune Operation**

An example is given by Figure9.



**Figure 9. Immunity Operation**

#### 4. Real-life Example

To test the effectiveness of our proposed method, it is applied to two real job shop schedule problems. In the experiments, the devices and process routes are firstly optimized according to the comprehensive fuzzy weight based dispatching rule method, and then the job shop schedule problems are solved by our developed fuzzy immune genetic algorithm.

##### 4.1. Case 1

Case 1 is a flexible job shop scheduling problem for a digital workshop, which could produce 5 kinds of workpieces. There are 10 Computer Numerical Control (CNC) machines, including 3 economic type I ones, 2 economic type II ones and 2 CNC centers. According to the real situation, 3 machines are operating other jobs, 2 are under repair, and 2 other ones enlarged their costs. Therefore, in order to get the maximum satisfaction rate or the minimum cost, not all devices need to be used. Firstly we optimize the machines according to the comprehensive fuzzy weight based dispatching rule method. The optimization procedure is shown as follows:

1. Define the set of Machines to be evaluated  $M=\{M_1, \dots, M_{10}\}$ , set the evaluation threshold value  $\lambda$  as 80.
2. Build the set of factors (objectives for optimization)  $F=\{f_1, f_2, f_3, f_4\}$ , where  $f_1, f_2, f_3$  and  $f_4$  are the ratio of vacancy processing time, cost rate, processing time, and criticality degree, respectively.
3. Construct the evaluation level set  $V=\{v_1, \dots, v_4\}=\{\text{complete, good, usable, unusable}\}$ .
4. Build the weight set. After establishing evaluation parameters, we assign corresponding weights according to relative importance between different layers and between different indexes by experience. For example, the final weight set of  $M_1$  is determined as  $\{0.2, 0.35, 0.1, 0.35\}$ .
5. Construct the comprehensive evaluation array by experience. For example, the component of  $M_1$  is set as

$$\left\{ \begin{array}{cccc} 0.3 & 0.45 & 0.15 & 0.1 \\ 0.28 & 0.42 & 0.2 & 0.1 \\ 0.4 & 0.35 & 0.15 & 0.1 \\ 0.38 & 0.32 & 0.18 & 0.18 \end{array} \right\}$$

6. Compute the comprehensive fuzzy weight vector according to Eq. (9). The component of  $M_1$  is computed as  $\{0.2, 0.35, 0.1, 0.18\}$ .



7. Calculate the final evaluation result vector according to Eq. (10). The final result is that  $E = \{ 80.78, 81.62, 87.85, 91.51, 82.34, 73.45, 78.32, 65.24, 71.57, 72.45 \}$ .

8. Remain the five devices with more than 80 evaluation values and delete others.

After the device optimization, we perform the fuzzy immune genetic algorithm to deal with the job shopping scheduling problem with selected five devices. The processing times are shown in Table 2. Then the processing time are fuzzified with triangle membership function, while the expected delivery date is described with the trapezoidal membership function. They are shown in Table 3. The fitness of the algorithm, namely the satisfaction ratio could be calculated according to Eq. (5).

To test the effectiveness of our proposed algorithm, four existing methods are applied to the problem for comparison. The first method is conventional genetic algorithm (GA), the second immune genetic algorithm (IGA), the third our proposed fuzzy immune genetic algorithm (FIGA) but without device optimization, and the last FIGA with device optimization. We investigate four metrics to measure the performance of algorithms, namely the ratio of vacancy processing time, processing time, longest delay time and maximum satisfaction rate. The results are shown in Table. 4.

**Table 2. Processing Time**

Jobs	Operations	Processing time of optional devices				
		M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>
J <sub>1</sub>	O <sub>11</sub>	3	4	—	—	6
	O <sub>12</sub>	-	4	1	3	—
	O <sub>13</sub>	4	2	—	3	3
J <sub>2</sub>	O <sub>21</sub>	2	—	4	—	2
	O <sub>22</sub>	—	5	3	7	6
	O <sub>23</sub>	—	5	2	2	9
J <sub>3</sub>	O <sub>31</sub>	7	6	—	3	—
	O <sub>32</sub>	3	5	6	3	1

**Table 3. Fuzzified Processing Time and Expected Delivery Date**

Jobs	Operations	Processing time of optional devices					Expected delivery date
		M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	
J <sub>1</sub>	O <sub>11</sub>	1,3,5	2,4,6	-	-	4,6,8	8,10,12,14
	O <sub>12</sub>	-	2,4,6	1,1,5	1,3,5	-	
	O <sub>13</sub>	2,4,6	1,2,3	-	1,3,5	1,3,5	
J <sub>2</sub>	O <sub>21</sub>	1,2,3	-	3,4,5	-	1,2,3	10,20,30,40
	O <sub>22</sub>	-	3,5,7	2,3,4	3,7,11	4,6,8	
	O <sub>23</sub>	-	4,6,8	1,2,3	1,2,3	6,8,10	
J <sub>3</sub>	O <sub>31</sub>	4,7,10	3,8,13	-	1,3,5	-	4,8,18,25
	O <sub>32</sub>	1,3,5	3,5,7	3,6,9	2,4,6	1,1,3	

**Table 4. Results Comparison**

	GA	IGA	IFGA	IFGA with Device Optimization
<b>Ratio of vacancy processing time (%)</b>	11.86	<b>10.21</b>	11.40	10.23
<b>Processing time(s)</b>	4538	4387	3834	<b>3324</b>
<b>Longest delay time(s)</b>	74.23	70.31	64.53	<b>54.34</b>
<b>Maximum satisfaction rate(%)</b>	41.31	43.24	48.97	<b>58.04</b>

From Table 4, it is easy to see that IFGA with Device Optimization performs the best on three metrics except for a little less than IGA on the ratio of vacancy processing time. And IFGA performs the second. It outperforms GA on all the four metrics and beats IGA on three of metrics, respectively. Evidently, among the four methods, GA is the worst performer. Compared IFGA and IFGA with Device Optimization, it can be seen that the latter is better than the former on all the four metrics.

#### 4.2. Case 2

Case 2 is a flexible job shop scheduling problem for a press line, which could produce 6 kinds of workpieces and includes 9 devices. There are 2 machines to operate other jobs, 3 under repair, and 3 enlarged their costs. Similarly as Case 1, we optimize the machines according to the comprehensive fuzzy weight based dispatching rule method, and remain four devices for processing. The processing time is shown in Table 5. The fuzzified processing time and expected delivery date are shown in Table 6. Comparison results are listed in Table.7.

**Table 5. Processing Time**

Job	Operations	Processing time of optional devices			
		M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>
<b>J<sub>1</sub></b>	O <sub>11</sub>	5	3	—	2
	O <sub>12</sub>	6	—	1	3
	O <sub>13</sub>	5	2	—	5
<b>J<sub>2</sub></b>	O <sub>21</sub>	3	—	6	—
	O <sub>22</sub>	2	4	-	9
<b>J<sub>3</sub></b>	O <sub>31</sub>	8	5	—	3
	O <sub>32</sub>	3	8	4	3
	O <sub>33</sub>	4	7	3	6
	O <sub>34</sub>	2	—	5	2
	O <sub>35</sub>	5	9	5	—

**Table 6. Fuzzification Result**

Job	Operations	Processing time of optional devices				Expected delivery date
		M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	
J <sub>1</sub>	O <sub>11</sub>	3,5,6	2,3,5	-	1,2,3	10,23,32, 42
	O <sub>12</sub>	4,6,8	-	1,1,2	2,3,5	
	O <sub>13</sub>	4,5,6	1,2,3	-	4,5,6	
J <sub>2</sub>	O <sub>21</sub>	2,3,4	-	3,5,7	-	7,12,18,2 4
	O <sub>22</sub>	1,2,3	2,4,6	-	3,7,1 1	
J <sub>3</sub>	O <sub>31</sub>	7,9,11	3,6,9	-	1,3,7	15,22,30, 40
	O <sub>32</sub>	1,4,7	3,6,10	2,4,6	2,3,4	
	O <sub>33</sub>	3,6,9	6,7,10	2,3,4	3,6,9	
	O <sub>34</sub>	1,2,3	-	3,5,7	1,2,3	
	O <sub>35</sub>	4,5,6	4,9,14	3,5,7	-	

**Table 7. Results Comparisons**

	GA	IGA	IFGA	IFGA with Device Optimization
<b>Ratio of vacancy processing time (%)</b>	12.05	<b>10.25</b>	11.34	10.31
<b>Processing time(s)</b>	45657	43576	34745	<b>31325</b>
<b>Longest delay time(s)</b>	184.32	187.36	165.57	<b>150.37</b>
<b>Maximum satisfaction rate(%)</b>	43.29	44.36	50.87	<b>59.91</b>

The results are much similar with those of Case 1. Among the four algorithms, IFGA with Device Optimization also performs the best on three metrics and IGA outperforms the others on the ratio of vacancy processing time. IFGA beats GA on all the four metrics and is better than IGA on three of metrics, respectively. About IFGA and IFGA with Device Optimization, one can see that the latter outperforms the former on all the four metrics.

#### 4.3. Analysis

From the experimental results on two real job shop scheduling problems, it can be seen that the proposed fuzzy immune genetic algorithm is better than the conventional genetic algorithm or immune genetic algorithm. Furthermore, if we conduct the device optimization in advance, the performance could be improved around ten percent. That is to say, both the proposed fuzzy immune genetic algorithm and the comprehensive fuzzy weight based dispatching rule method for device optimization are effective.

### 5. Conclusions

In this paper, we firstly develop a comprehensive fuzzy weight based dispatching rule method to evaluate the fuzzy weights of devices and then to optimize the machines for scheduling. Next a fuzzy immune genetic algorithm is proposed to deal with the scheduling problems. In the method, we use the fuzzified processing time and delivery date to evaluate a solution by so called maximum average satisfaction

rate. The experimental results on two real job scheduling problems show the effectiveness of our proposed method.

## Acknowledgements

The authors are grateful to the support of the National Natural Science Foundation of China (61272207, 61373050, 61202306, 61472049 and 61402193) and the Science Technology Development Project from Jilin Province of China (20130206003SF, 20130101070JC, 20100507).

## References

- [1] Kashan, Ali Husseinzadeh, B. Karimi, and F. Jolai, "An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes." *Engineering Applications of Artificial Intelligence*, vol.23, no.6, pp.911-922, (2010).
- [2] Branke, Juergen, and C. W. Pickardt, "Evolutionary search for difficult problem instances to support the design of job shop dispatching rules." *European Journal of Operational Research*, vol.212, no.1, pp.22-32, (2011).
- [3] Mouelhi-Chibani, Wiem, and H. Pierrel, "Training a neural network to select dispatching rules in real time." *Computers & Industrial Engineering*, vol.58, no.2, pp. 249-256, (2010).
- [4] Olafsson, Sigurdur, and X. Li, "Learning effective new single machine dispatching rules from optimal scheduling data." *International Journal of Production Economics*, vol.128, no.1, pp. 118-126, (2010).
- [5] A. P. Cucala, A. Fernández, C. Sicre, and M. Domínguez, "Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertain delays and driver's behavioral response." *Engineering Applications of Artificial Intelligence*, vol.25, no.8, pp.1548-1557, (2012).
- [6] Giannopoulos, Nikos, V. C. Moulianitis, and A. C. Nearchou, "Multi-objective optimization with fuzzy measures and its application to flow-shop scheduling." *Engineering Applications of Artificial Intelligence*, vol.25, no.7, pp.1381-1394, (2012).
- [7] J. P. Wang, H. J. Zhu, and J. Zhou, "Optimal sub-population genetic algorithm for flexible flow shop scheduling problem." *Application Research of Computers*, vol.29, no.2, pp.442-441, (2012).
- [8] Baykasoğlu, Adil, and L. Özbakır, "Analyzing the effect of dispatching rules on the scheduling performance through grammar based flexible scheduling system." *International Journal of Production Economics*, vol.124, no.2, pp.369-381, (2010).
- [9] Saidi-Mehrabad, Mohammad, and P. Fattahi, "Flexible job shop scheduling with tabu search algorithms." *International Journal of Advanced Manufacturing Technology*, vol.32, no.5-6, pp.563-570, (2007).
- [10] Y. C. Ho, and H. C. Liu, "A simulation study on the performance of pickup-dispatching rules for multiple-load AGVs." *Computers & Industrial Engineering*, vol.51, no.3, pp.445-463, (2006).
- [11] X. Wang, L. Gao, C. Zhang, and X. Shao, "A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem." *International Journal of Advanced Manufacturing Technology*, vol.51, no.5-8, pp.757-767, (2010).
- [12] K. Kianfar, S. M. T. F. Ghomi, and B. Karimi, "New dispatching rules to minimize rejection and tardiness costs in a dynamic flexible flow shop." *International Journal of Advanced Manufacturing Technology*, vol.45, no.7-8, pp.759-771, (2009).
- [13] W. L. Wang, L. I. Jun fang, and J. Wang, "Double-population genetic algorithm for multi-objective Job Shop scheduling problem." *Computer Integrated Manufacturing Systems*, vol.17, no.4, pp.808-815, (2011).
- [14] H. J. Lee, D. Y. Park, and B. S. Ahn, "A fuzzy expert system for the integrated fault diagnosis." *IEEE Transactions on Power Delivery*, vol.15, no.2, pp.833-838, (2000).
- [15] Yazgan, Harun Resit, "Selection of dispatching rules with fuzzy ANP approach." *International Journal of Advanced Manufacturing Technology*, vol.52, no.5-8, pp.651-668, (2011).
- [16] D. Y. Ma, Y. C. Liang, X. S. Zhao, Z. X. Li, and X. H. Shi, "The Application of Fuzzy System Group in Intelligent Diagnosis for Power Transformer." *2012 Third International Conference on Digital Manufacturing & Automation IEEE*, pp.1206-1209, (2011).
- [17] Z. Q. Geng, Y. R. Zou, "Study on Job Shop Fuzzy Scheduling Problem Based on Genetic Algorithm." *Computer Integrated Manufacturing Systems*, vol.8, no.8, pp.616-620, (2002).
- [18] Vázquez-Rodríguez, José Antonio, and S. Petrovic., "A new dispatching rule based genetic algorithm for the multi-objective job shop problem." *Journal of Heuristics*, vol.16, no.6, pp.771-793, (2010).
- [19] Y. S. Chai, S. D. Sun, J. J. Yu, X. L. Wu, "Job shop dynamic scheduling problem based on immune genetic algorithm." *Journal of Mechanical Engineering*, vol.41, no.10, (2005).
- [20] G. Zhang, X. Shao, P. Li, and L. Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem." *Computers & Industrial Engineering*, vol.56, no.4, pp.1309-1318, (2009).

## Authors



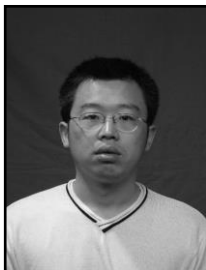
**Deyin Ma**, PhD, mainly engaged in the research of intelligent computing. The five papers as the first author, the SCI retrieval 1 papers (Engineering Applications of Artificial Intelligence, the impact factor of 1.665), 2 papers EI core journals (in publish), international conference on EI 2 papers.



**Xuming Han**, he is a Professor in Changchun University of Technology, He received M.S. degree and Ph.D degree in the College of Computer Science and Technology, Jilin University and Tianjin University, respectively. He has published more than 40 journal papers and conference papers. His research interests include data mining, swarm intelligence and computation intelligence, etc. He has received 2 Projects supported by the National Natural Science Foundation of China in recent years, 2 Scientific and Technological Projects of Jilin Province, etc..



**Yanchun Liang**, he was born in 1953. He is a professor in the College of Computer Science and Technology, Jilin University, China. His research interests include Intelligent Computation and Bioinformatics.



**Xiaohu Shi**, he was born in 1974. He is a professor in the College of Computer Science and Technology, Jilin University, China. His research interests include Machine Learning and Bioinformatics.

