# An Immune Particle Swarm Optimization Method for Permutation Flow Shop Scheduling Problem

LIN WANG, JIANHUA QU* ,YUCHENG JIA
School of Management Science and Engineering University
Shandong Normal University
Jinan, Shandong province
CHINA
{moanfly, qujh1978, jiayucheng2014}@163.com

*Abstract:* - Permutation Flow Shop Scheduling Problem (PFSP) is a complex combinatorial optimization problem with strong engineering background. To solve the PFSP with makespan criterions, an immune particle swarm optimization (IPSO) algorithm was proposed. The initial solution of the algorithm is generated by the famous heuristic NEH algorithm, it was used to initialize the particle of global extreme values. Then we add a Dynamic Disturbance Term (DDT) in the velocity updating formulation of the particle, it used to prevent optimizing course from trapping the local minimum. Density and Immune Selection mechanism of Immune algorithm (IA) are used in the iterative process to select the optimal particle through the choice probability equation. The vaccination and memory operation to guide the global optimization process. At last, computational results show that the IPSO algorithm is effective robust and has a high performance.

*Key-Words:* - Permutation Flow Shop Scheduling; Particle Swarm Optimization Algorithm; Immune Algorithm; Makespan

## 1 Introduction

Production scheduling plays an important role in manufacturing systems. The technology for solving and optimization of scheduling problem is the foundation and key to realize the rationality, automation and integration for the production process, and already be widely used in many computer system, transport dispatching and production management[1]. It is very important for the manufacturing enterprise to develop effective and efficient and practical scheduling technologies and approaches.

Flow shop scheduling is a typical of widely studied scheduling problems which has been proved to be an NP-hard problem in the sense of its high computational complexity[2]. In this problem, there are n jobs and m machines, the job sequence on each machine is the same. And FSP dealing with n jobs on m given machines has been one of the considerable researches, various approaches to this problem has been proposed after the publication of Johnson's work[3].

FSP contains many complexity Scheduling problems. The problem of scheduling a set of jobs on a set of machines in a permutation flow shop scheduling problem is an important issue in the field of operation research. It was built on the foundation of the PFSP, and demands the processing sequences of all jobs are the same on each machine. also the PFSP is a kind of combinatorial optimization problem, which has wide application background.

In recent years, many researcher have proposed many novel algorithms both in mathematical field an computing intelligence field for solving the scheduling problem. He Long-min elaborate a model that the hybrid two-stage flexible flow shop scheduling problem with m identical parallel machines [4]. Chuyang Wang put forward to a fast iterated local search algorithm which based on high order neighborhood to solve no-wait flowshop scheduling problem [5]. Gui zhe proposed a hybrid discrete harmony search algorithm for flow shop scheduling problem [6].Yu Xue proposed a self-adaptive learning based discrete evolution algorithm for solving this problem [7]. As the emergence of new techniques from the field of artificial intelligence, much attention has been devoted to some relatively current evolutionary computation techniques such as Genetic algorithms (GA) [8]; Simulated Annealing algorithms (SA) [9]; Particle Swarm Optimization algorithms (PSO) [10]; Colony algorithm (ACA) [11] have been widely applied to PFSP and proven to be useful optimization techniques .

Particle Swarm Optimization (PSO) algorithm is an evolutionary computation technology developed by Eberhart and Kennedy [12]. It is inspired by the emergent motion of a flock of bird of fish searching for food. The basic idea was to find optimal solutions through the collaboration and sharing of information among individuals in the group. PSO algorithm has been used to solve a variety of difficult optimization problems, including engineering applications and optimization problem [13]. Compared with other algorithms, the convergence process of the standard versions of the above algorithms tend to be slow to some extent for practical-scale problem. Motivated by this perspective. PSO algorithm has been widely used with great success in providing approximate solutions to many scheduling problems [14].

In this paper, we proposed an effective immune PSO algorithm for solving the PFSP to minimized makespan. A sequence of job was generated by heuristics NEH algorithm ,this method produce the global particle extreme at the initialization. And we add a dynamic disturbance term in the velocity updating formulation of the particle, it used to prevent optimizing course from trapping the local minimum. The Immune algorithm (IA) was introduced in the particle population, the density and immune selection operations based on the IA can maintain the individual diversity, this method can selected the suitable particle into next generation through the choice probability equation. The memory and vaccination mechanism for particles can compensates the shortcoming of easily falling into the local optimum. At last, a numerical example is given to show the feasibility and the effectiveness of the proposed IPSO algorithm.

The remaining contents of this paper is organized as follows: The basic concepts of the PFSP and PSO algorithm are described in Section 2. In Section 3, the implementation of IPSO algorithm is explained in detail. Section 4 gives a description of IPSO algorithm. Computational results and comparisons are listed in Section 5. Finally, we end the paper with some conclusions and future work in Section 6.

# 2 Problem Formulation

## 2.1 mathematical description of simple-objective PFSP

This paper is inspired to solve the PFSP in a manufacturing environment, and this scheduling belongs to a static scheduling problem. In order to simplify the mathematical model, we have to make a assumption, once the processing of an operation starts, it cannot be interrupted at any time until the job has been released by the machine [15]. For an example of PFSP, a set of n jobs (denoted by $J = \{1,2 \dots, n\}$) are waiting to be processed on a set of m machines (denoted by $M = \{1,2, \dots, m\}$ ). Some basic assumptions of this problem are as follows:

1. n jobs are processed according to the same technology line on m machines;
2. The processing sequence of all jobs is the same on each of m machines.
3. Each same job can only be processed on one machine at the same time, and one machine can process only one job at a time;
4. There is no priority among jobs and no machine breakdown during jobs process.
5. The processing time of every job on each machine is known to us;
6. Every job is independent with each other, and the preparative time of each job is zero;
7. All the machines are continuously available throughout the production process;
8. The transportation time to deliver jobs between two different machines is neglected, and the setup time for machines to switch between two different jobs is neglected:

The problem is to decide a schedule scheme for the jobs on the machines to minimize the makespan, it was the maximum of completion times needed for processing all the jobs. The makespan of a scheduling is equal to the total processing times of the operations on a critical path from the first operation of the first job to the last operation of the last job. The mathematical model of the problem is formulated as follows:

$$C(1,1) = t_{11}; \quad (1)$$

$$C(i,1) = C(i-1,1) + t_{i1}, \ (i = 2,3,4 \dots, n) \quad (2)$$

$$C(1,j) = C(i,j-1) + t_{1j}, \ (j = 2,3,4 \dots, m); \quad (3)$$

$$C(i,j) = \max[C(i-1,j), C(i,j-1)] + t_{ij}; \\ (i = 2,3,4 \dots, n, \ j = 2,3,4 \dots, m); \quad (4)$$

$$f: \min[\max C(i,j)]; \quad (5)$$

$$C(i,j) \geq 0, (i = 1,2,3 \dots n, \ j = 1,2,3 \dots, m)\ ; \quad (6)$$

The objective function (5) is to minimize makespan that is the completion time of the last operation on a critical path. Constraint (6) forces the finish times to be nonnegative. The PFSP is to find a permutation $\pi^*$ ($\pi^*$ is a sequence of jobs) in the set of all permutation scheduling $\prod$ . Such that. $C_{max}(\pi^*) \leq C_{max}(\pi), \forall\, \pi \in \prod$

## 2.2 The standard PSO algorithm

Particle swarm optimization (PSO) [12] algorithm is an evolutionary computation technique which inspired by social behavior of bird flocking or fish schooling. It used the physical movements of the individual in the swarm and has a flexible and well-balanced mechanism to enhance and adapt to the global and local exploration abilities. In the system, each potential solution call particles, which flies in the $D-$dimensional problem space with a velocity. These particles dynamically adjusted the position according to the flying experience of its own and other particles.

The algorithm firstly initializes particles randomly in the multidimensional space and speed space, each particle has a position vector $x_i$, a velocity vector $v_i$, Then it evaluates each particle according to the fintess function, the position at which the best fitness $p_{ij}(t)$ was encountered by the particle and the position of the best particle $p_{gj}(t)$ in the swarm. In each generation particle with the best solution shares its position coordinates information with the rest swarm. the new position and velocity of each particle i is determined by the following equations:

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 * r_1 * \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 * r_2 * \left(p_{gj}(t) - x_{ij}(t)\right); \quad (7)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1); \quad (8)$$

The parameter $w$ is called the inertia weight which determines the influence of the old velocity. $c_1$ and $c_2$ are positive acceleration constant parameters called acceleration coefficients which control the maximum step size. $t$ is iteration counter. $r_1(t)$ and $r_2(t)$ are the are two independently uniformly, distributed random variables with range positive constants.. The termination criterion for the iterations is determined according to whether the max generation or a designated values of the fitness of global particle is reached[16].

# 3 An IPSO algorithm description

## 3.1 Initialized the optimal position of the particle

In the PSO algorithm, the initial solution has an important effect on the performance of the algorithm. Initial solution as starting point of their stochastic solution improvements schemes which has attracted much attention for the optimized initial solution. In recent years, many constructive heuristic algorithms has been used in the initial solution, various companion researches concluded that NEH algorithm is the most efficient to find the initial solution for the PFSP. The main steps of NEH algorithm are listed as follows:

1 Order the jobs by the non-increasing sums of processing times on the machines;

2 Take the first two jobs and schedule them in order to minimize the partial makespan as if there were only these two jobs;

3 For $k = 3$ to n do 4;

4 Insert the $k$-th job at the place, which minimizes the partial makespan among the $k$ possible ones [17].

There is a great influence for the efficiency of the algorithm to seek the global optimum that the individual of the initial population is good or bad. so we take the global extreme which generated by the heuristics NEH algorithms. For the NEH algorithm, we can only get a sequence of jobs, it was not the continuous number which used in PSO algorithm. So we adopted the transformation method to switch the job sequence to the position vector in certain space, the transformation method between the NEH algorithm individual expressed as a real number and the job sequence of the PFSP switch to particle position as follows:

$$X_{NEH,j} = \left(S_{NEH,j} - 1 + R\right)/n; \quad (j = 1,2,3 \cdots n) \quad (9)$$

$x_{NEH,j}$: $j$-dimensional value of particle position; $s_{NEH,j}$: j-dimension solution which was produced by the NEH algorithm; $n$ was the number of the jobs; R: produced by the uniform random on the interval discrete set of values which was in the range of $(0,1)$.

## 3.2 The Dynamic Disturbance Term

PSO algorithm is an evolutionary algorithm through the cooperation and competition among the individuals in the swarm. Through this mathematical model, the search space is dependent on the knowledge of history and the global particle. As the particle moves more closer to the global extreme, the diversity of population began to disappear, once the velocity of the particle is zero, the algorithm get into local optimal value.

Coping with such disadvantages of PSO algorithm being easy to run into local extreme. we proposed a method based on DDT and changed the speed formula that the particle can jump out local optimum. A successively decreasing disturbance term was added into the velocity updating formula when the particle position updating was too slow or kept relatively unchanged in the middle and final evolutions periods. This method take the following formula:

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 * r_1 * \left(p_{ij}(t) - x_{ij}(t)\right) + c_2 * r_2 * \left(p_{gj}(t) - x_{ij}(t)\right) + M; \quad (10)$$

$$M = k \times \left(\frac{t}{T_{max}} - 0.5\right); \quad (11)$$

K is the accommodation coefficient which the value between [0,1]. The formula which used in the location of the particle stay the same. And the framework of PSO basic velocity formula is analyzed, from the M, when the velocity of particles begin decreasing, this term can promised the speed of the particle not be zero. With the increase of the number of iterations, the fourth term will be increase gradually, it provides the guarantee for the algorithm to jump out of the local minimum [18].

### 3.3 Immune algorithm (IA)

Immune algorithm (IA) is one of the approximate optimization algorithm. It was built by taking advantage of physiology immune system. The vertebrate immune system comprises the innate and adaptive defense mechanisms which provides the host body with a means of protection against infectious agents. The aim of leading immune concepts and methods into original optimal algorithm is theoretical to utilize the local characteristic information for seeking the ways and the means of finding the optimal solution when dealing with difficult problems [19].

Based on the mechanism of biological organism's immune system. such as diversity, concentration adjustment, learning, memory and self-adaptive control, we introducing the IA in the PSO algorithm, which based on the concentration of immune selection and density mechanism to ensure the diversity of the particle swarm. The initial particle swarm as viewed the antibody population, the memory cell follow the global extreme when the antibody are renewed. The vaccine of the algorithm is obtained from the global solution iterated N times, and refreshed with changing of the optimal values of the different generations. The procedure of the IPSO approach is discussed below:

(1) Generation of initial antibody population.

The initial antibody population is generated from the initial particle swarm. In the particle swarm ,we can obtained the excellent antibodies form the global particle Which stored in the memory cells.

(2) Proliferation of antibody population.

In this process, to generate some antibodies randomly to improve the diversity of antibodies population. M is the number of the new antibodies. After that, the new population with $N + M$ antibodies.

(3) Selection of the antibody population.

We take N antibodies from the new antibody population by some rules. In this paper, we take a self-adjust method by the density and affinity in the immune system, in the antibody population, the density of each antibody can calculated by the formula:

$$D(x_i) = \frac{1}{\sum_{j=1}^{N+M}|f(x_i) - f(x_j)|}; \; i = 1,2,3,\cdots N+M \quad (12)$$

The selection probability produced by the density of each antibody:

$$P_1(x_i) = \frac{\frac{1}{D(x_i)}}{\sum_{i=1}^{N+M}\left(\frac{1}{D(x_i)}\right)} = \frac{\sum_{j=1}^{N+M}|f(x_i) - f(x_j)|}{\sum_{i=1}^{N+M}\sum_{j=1}^{N+M}|f(x_i) - f(x_j)|};$$

$$i = 1,2,3 \cdots N+M; \quad (13)$$

Certainly, in order to take the choose operation in each antibody more fairly and accurately, we consider another method which based on the affinity of each antibody:

$$Q(x_i) = \frac{1}{f(x_i)}; i = 1,2,3,\cdots N+M \quad (14)$$

The selection probability produced by the affinity of each antibody:

$$P_2(x_i) = \frac{Q(x_i)}{\left(\sum_{j=1}^{N+M} Q(x_j)\right)}; \; i = 1,2,3,\cdots N+M \quad (15)$$

The selection probability of each antibody:

$$P(x_i) = P_1(x_i) + P_2(x_i); \; i = 1,2,3,\cdots N+M; \quad (16)$$

According to the formula (13) , the choice opportunity of each antibody can be used to sort from highest to lowest. we take the N number in the top of antibody population .

(4) Vaccination of antibody population

To take an antibody from the new antibody population randomly, and take an antibody from the memory cells randomly. And we can calculate the fitness values of these antibodies, if the antibody in the memory cells was better than the antibody in the population, it has replaced the original optimal antibody, otherwise it will reserved the antibody in the population until the better antibody emerged in the later iterative process.

# 4 The IPSO Algorithm for PFSP

## 4.1 The design of p article encoding and decoding

In the PFSP, the important issue for applying PSO algorithm to PFSP is how to encode a schedule to a search solution. From this, we set up a search space of n dimensions for a problem of n jobs. And took the smallest position value (SPV) [20] method as the encoding approach to solving this problem.

The location of particle was described n dimension vector, the sequence of the particles' location represented the corresponding processing sequence of jobs. The most important thing is the decoding approach which can be described as below: we taking the real number on the position of particle from small to large order, and establish a mapping between the particle location and dimension, the dimension j was mapping with the sequence of the job, then we can get a processing sequence of the job. Certainly, each dimension of particle was natural numbers which was no longer than the amount of jobs. We give a Tab 1 to explain this encoding and decoding method:

Table 1. An example of SPV

| Dimension $j$ | Position $x_{ij}$ | The sequence of the job |
|---|---|---|
| 1 | 1.80 | 5 |
| 2 | -0.99 | 2 |
| 3 | 3.01 | 4 |
| 4 | -0.72 | 1 |
| 5 | -1.2 | 6 |
| 6 | 2.15 | 3 |

## 4.2 Setting parameters

The inertia weight w is an important control parameters to search ability of the IPSO algorithm, A large inertia weight facilitates searching new area while a small inertia weight facilitates fine-searching in the current search. Suitable selection of the inertia weight provides a balance between global exploration and local exploitation, which have an obvious effect on the convergence of the algorithm and the exploration integrality.

Given this, we take the approach on the basic of the characteristic of the PFSP. The method of linear inertia weight method was be used. Linearly decreasing the inertia weight from a relatively large value to a relatively small value through the course of IPSO running, it can have a wide range of solution space on global searching to find the best solution at the beginning of the PSO algorithm process, and to avoid the global extreme shocked in the global searching process at the end of the PSO algorithm process. The parameter w is calculated by the formula:

$$w(t) = w_{max} - (w_{max} - w_{min}) * (t/t^*) \qquad (17)$$

$w_{min}, w_{max}$: the minimum and maximum of the inertia weight. $t$ : current iteration or generation number. $t^*$ : the total number of iterations .

## 4.3 Fitness function

Fitness is used as performance evaluation of particles in the swarm. In the IPSO algorithm, the objective of PFSP with makespan criterion is to minimize the max complete time of all jobs, Particle with lower fitness will be superior to other particles and should be reserved in the search process. The makespan of the scheduling is equal to the total processing times of the operations on a critical path from the first operation of the first job to the last operation of the last job. So the fitness values of the individual is regarded as the complete time of the corresponding job permutation.

Step1: The individual is converted into the job permutation based on the SPV rule.

Step2: According to the formulas from (1) to (6) , calculate the processing time to get the makespan, Which is the fitness values of the individual.

## 4.4 The Procedure of IPSO algorithm

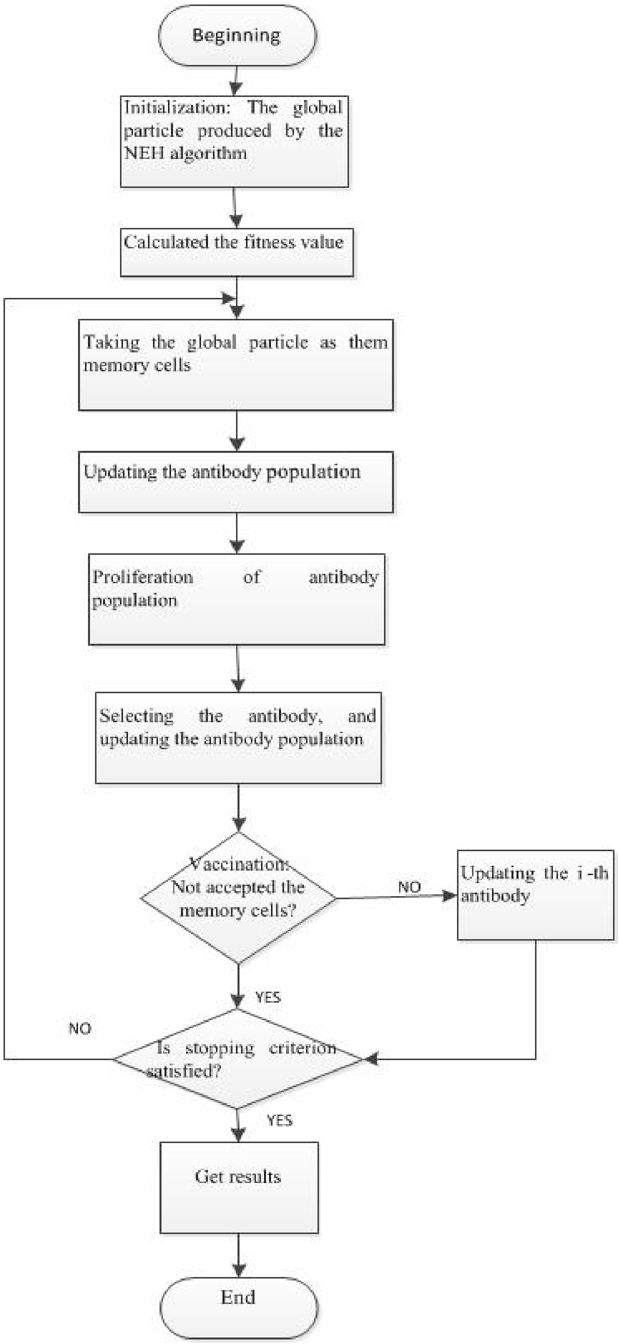



**Fig.1**.Flow chart showing basic steps in the IPSO algorithm

## 5 Simulation Results

In the section 5, we have introduced the IPSO algorithm. In order to evaluate the performance of the proposed IPSO algorithm. Some experiments are carried out. we take an example which is Car and Rec [21] benchmark problem to find the appropriate values, the previous two of the benchmark problem was used to test, such as Car1,Car2, Rec01, Rec03, and we compared the IPSO algorithm to the traditional PSO algorithm to make the simulation results signifinitly.

The machine is DELL N4020/P4 Intel 2.30 GHz/2GB. For the hybrid PSO algorithm which we have proposed, a swarm population size of 50 is used for the text problem. Default values for the parameters $c_1$ and $c_2$ have been used: $c_1 = c_2 = 2$ .The minimum of the inertia weight: $w_{min} = 0.4$ ,The maximum of the inertia weight: $w_{max} = 1.2$. And the immune antibody population for the parameter of M : $M = 30$ .And the maximum of generation: $t_{max} = 50$, and the particle swarm population : $N = 50$ . The algorithm have been executed in 3 independent runs to eliminate the stochastic influence.

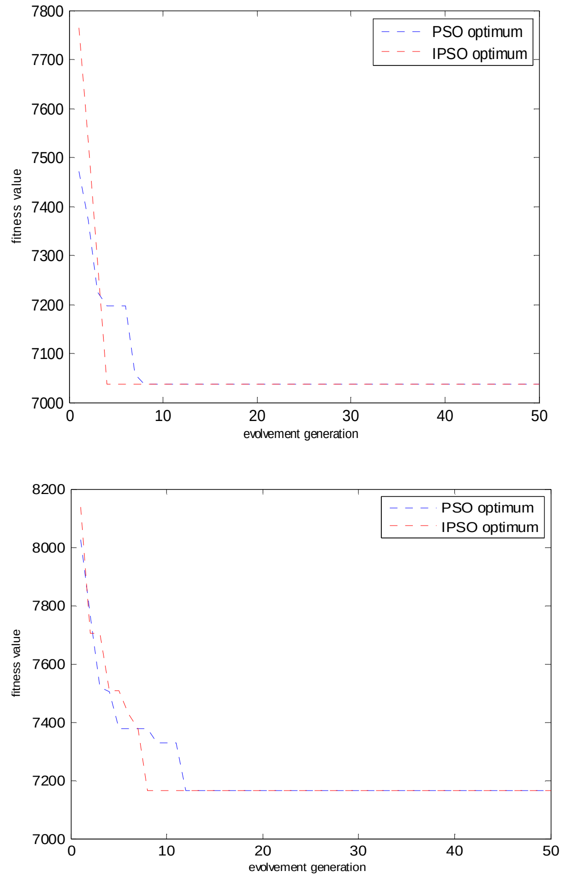
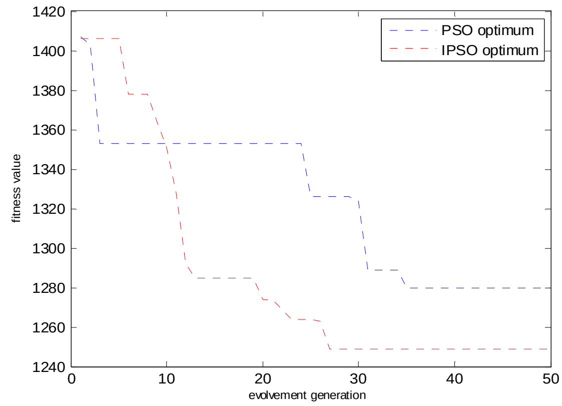



**Fig**.2.Convergence of the IPSO and PSO algorithm(Car1,Car2)



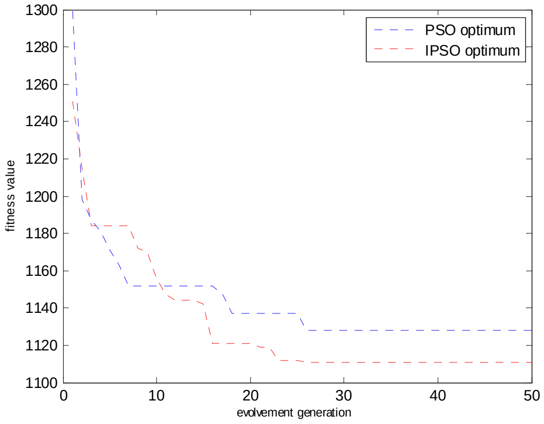



**Fig**.3.Convergence of the IPSO and PSO algorithm(Rec01,Rec03)

From Fig.2 and Fig.3. It was shown that the effectiveness of the IPSO algorithm for solving the PFSP, and has the better performance than the traditional PSO algorithm. In order to further verify the effectiveness of the proposed algorithm, we compared the results of the PSO algorithm and IPSO algorithm for the Taillard001 (20x5) [22] example.

Table 2 The processing time of the Ta001
(the measuring unit for minutes)

| m / n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 54 | 83 | 15 | 71 | 77 | 36 | 53 | 38 | 27 | 87 |
| 2 | 79 | 3 | 11 | 99 | 56 | 70 | 99 | 60 | 5 | 56 |
| 3 | 16 | 89 | 149 | 15 | 89 | 45 | 60 | 23 | 57 | 64 |
| 4 | 66 | 58 | 31 | 68 | 78 | 91 | 13 | 59 | 49 | 85 |
| 5 | 58 | 56 | 20 | 85 | 53 | 35 | 53 | 41 | 69 | 13 |

| m / n | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 76 | 91 | 14 | 29 | 12 | 77 | 32 | 87 | 68 | 94 |
| 2 | 3 | 61 | 73 | 75 | 47 | 14 | 21 | 86 | 5 | 77 |
| 3 | 7 | 1 | 63 | 41 | 63 | 47 | 26 | 75 | 77 | 40 |
| 4 | 85 | 9 | 39 | 41 | 56 | 40 | 54 | 77 | 51 | 31 |
| 5 | 86 | 72 | 8 | 49 | 47 | 87 | 58 | 18 | 68 | 28 |

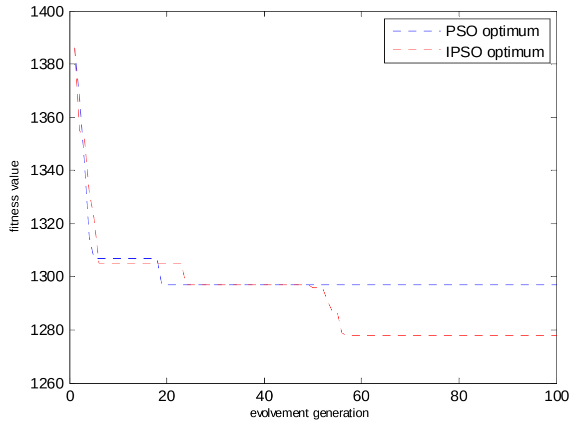



**Fig**.4.Convergence of the IPSO and PSO algorithm(Ta001)

Table3 Best Scheduling Results of PSO Algorithm

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1297 | 1297 | 1305 | 1297 | 1297 | 1297 | 1297 | 1305 | 1297 | 1303 |
| 1297 | 1297 | 1297 | 1297 | 1297 | 1297 | 1297 | 1297 | 1297 | 1301 |
| 1297 | 1297 | 1297 | 1297 | 1297 | 1305 | 1297 | 1301 | 1297 | 1297 |
| 1297 | 1305 | 1297 | 1305 | 1297 | 1314 | 1297 | 1297 | 1303 | 1297 |
| 1297 | 1297 | 1297 | 1298 | 1297 | 1305 | 1307 | 1297 | 1297 | 1297 |

Table4 Best Scheduling Results of IPSO Algorithm

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 1297 | 1297 | 1297 | 1297 | 1297 | 1297 | 1324 | 1293 | 1297 | 1297 |
| 1297 | 1297 | 1297 | 1297 | 1302 | 1297 | 1297 | 1297 | 1297 | 1278 |
| 1297 | 1297 | 1297 | 1302 | 1297 | 1297 | 1297 | 1297 | 1297 | 1297 |
| 1297 | 1297 | 1295 | 1297 | 1318 | 1297 | 1297 | 1297 | 1297 | 1297 |
| 1297 | 1297 | 1321 | 1297 | 1297 | 1297 | 1297 | 1297 | 1297 | 1310 |

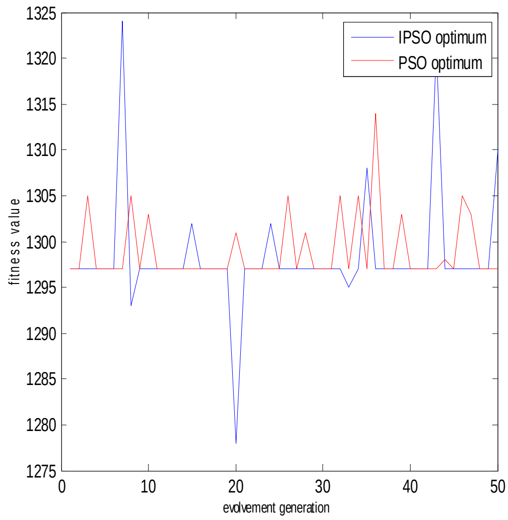



**Fig**.5. The results of the IPSO and PSO algorithm(Ta001)

Table 5 comparison of algorithms on Taillard001 (20 x 5)

| **Item** | **Optimal value** | **Worst value** | **Aver value** | **std** |
|---|---|---|---|---|
| **PSO** | 1297 | 1314 | 1298.84 | 3.6273 |
| **IPSO** | 1278 | 1324 | 1298.40 | 6.3149 |

From these specific examples, it shows the IPSO algorithm can find the best solution through the benchmarks in the PFSP. From the Fig.4 Fig.5 and Table 3, Table 4, Table 5, compared with PSO algorithm, it can find that the IPSO has a strong global search ability and not easy trapped into local extreme. All of this can be conducted that the IPSO algorithm is superior to the PSO algorithm with the same time requirements for the PFSP to minimized makespan.

Table6 Results of simulation tests for four algorithms

| Problem | Scale | | $C^*$ | NEH [23] | PSO | GA [24] | IPSO |
|---|---|---|---|---|---|---|---|
| | n | m | | | | | |
| **Car1** | 11 | 5 | 7038 | 7038 | 7038 | 7038 | 7038 |
| **Car2** | 13 | 4 | 7166 | 7376 | 7166 | 7166 | 7166 |
| **Rec 01** | 20 | 5 | 1247 | 1303 | 1247 | 1247 | 1247 |
| **Rec 03** | 20 | 5 | 1109 | 1182 | 1109 | 1109 | 1109 |
| **Ta 001** | 20 | 5 | 1278 | - | 1297 | 1297 | 1278 |
| **Ta 005** | 20 | 5 | 1235 | - | 1250 | 1250 | 1235 |

At last, from the Table 6, this four algorithms can also find the optimal makespan for solving the classical examples. And the IPSO algorithm has a better performance than other three algorithms. From the experiments, it is possible to see the performance of IPSO algorithm is improved by incorporating the heuristic algorithm and Immune method in PSO algorithm, these improvements are more benefit for solving the scheduling problem.

# 6 Conclusion

In this paper, we proposed the Immune particle swarm optimization approach to minimized makespan for PFSP. Some simulation results proved the validity of the IPSO algorithm in solving this problem. The heuristics NEH algorithm produce the global particle extreme in the initialization process. A dynamic disturbance term can prevent optimizing course from trapping the local minimum, which be used in the velocity updating formulation of the particle. The features of IA is special mechanisms which include the ability of maintaining diversity of antibodies, it can selected the optimal particle to guide population optimization process. The density and memory mechanism to enhance the ability for seeking the global extreme. For the vaccination and immune selection operation, it can compensates the shortcoming of easily falling into the local optimum, and also improved the performance ability of PSO algorithm. Through the above research, the future work is to investigate how to combine the intelligence knowledge of various problem with some principle of particle swarm system, and to develop other PSO-based for PFSP and other scheduling problems.

**Acknowledgment**

This work was supported by: Science-Technology Program of the Higher Education Institutions of Shandong Province, China (No.J12LN22) and Research Award Foundation for Outstanding Young scientists of Shandong Province, China (No.BS2012DX041) and the National Natural Science Foundation of China (Grant No.61472231).

*References:*

[1]Yun JING,Shiwei HE,Zhiqiang TIan,Qi SONG,Quantum Immune Clonal Algorithm for NO-Idle Flow Shop Scheduling Problem.Systerm Simulation,Technology&application.No.14,2009,pp. 209-212.

[2]J. K. Lenstra, A. H. G. R. Kan, and P. Brucker, “Complexity of machine scheduling problems,” Annals of Discrete Mathematics, No. 1,1997, pp. 343–362, .

[3]ChaochaoWu,.XingshengGu.,A Genetic Algorithm for Flow shop Scheduling with Fuzzy processing Time and Due Date[J],proceedings of 5th World congress on control and Automation,June,2004:pp.2983-2942.

[4]HE Long-min,SUN Shi-jie,A hybrid two-stage flexible flowshop scheduling problem with m identical parallel machines and a burn-in processor separately[J].Joutnal of Shanghai University(EnglishEdition),Vol02,No.28,2007,pp.33 -38.

[5]Chuyang, Wang.Fast Iterated Local Search Algorithm with High Order Neighborhood for No-wait Flowshop Scheduling Problem[A]. Proceedings of the 2014 Chinese Control and Decision Conference (CCDC),2014,pp.4461-4465.

[6]CUI Zhe, GU Xing-sheng, Hybrid Discrete Harmony Search Algorithm for Flow Shop Scheduling with Limited Buffers[J].Journal of Donghua University (English Edition)vol04,No.30,2014,pp.165-170.

[7]Yu Xue,Yi Zhuang,Tianquan Ni. Self-adaptive learning based discrete differential evolution algorithm for sloving CJWTA problem[ J].Journal of systerms Enigeering and Electronics.Vol02,No.15,2014,:59-68.

[8]Wenbin Gu,Dunbing Tang,Lei Zheng. Solving Job-Shop Scheduling Problem Based on Improved Adaptive Particle Swarm Optimization Algorithm[J].Transactions of Nanjing University of Aeronautics and Astronautics, Vol.31,No.5,2014,pp.:559-567.

[9]Gao YE , Xue RUI, An Improved Simulated Annealing Algorithm for TSP [A].Proceeding of 2013 5th IEEE International Conference On Broadband Network & Multimedia technology[C].2013.

[10]Jianping Dou ,Jun Li .A discrete Particle Swarm Algorithm foe Makespan Minimization in the Permutation Flowsshop Scheduling Problem[J],proc IEEE international conference on evolutionary computation,2013,pp.69-73.

[11]Zhang peng, Wang ling. Improved Fruit Fly Optimization Algorithm for Solving Lot-Streaming Flow-Shop Scheduling Problem[J].Journal of Donghua University (English Edition),Vol31,No.02,2014,pp.165-170.

[12]Eberhart R C.Kennedy J.A New Optimizier Using Particle Swarm Theory [C].Processing of the sixth International Symposium on Micro Machine and Human Science, Micro Machine and Human Science, Nagoya Japan,1995,pp.39-43.

[13]Jianjun Qi,Bo Guo,Hontao Lei,Tao Zhang.Solving resource availability cost problem in project scheduling by pseudo particle swarm optimization[J].Journal of Systems Engineering and Electronics,Vol25,No.01,2014,pp.69-76.

[14]Hai-bo Tang, Chun-ming YE.Permutation Flow Shop Scheduling Algorithm based on a Hybrid Particle Swarm Optimization[A].Proceedings of 2010 IEEE the 17th International Conference on Industrial Engineering and Engineering Management[C].2010.

[15]M.Tasgetiren, M.Sevkli,Y.C.Liang and G.Gencyilmaz, "Paricle swarm optimization algorithm for permutation flow shop sequencing problem."Ant Colony Optimization and Swarm Intelligence,2004,pp.366-385..

[16]LIIU Bo,WANGLing,JINYIhui.An Effective Hybrid PSO-based Algorithm for Flow Shop Scheduling to Minimize Makespan[J].Chaos,solitions&fractals,Vol35,No5,2008,pp.851-861.

[17]WANG Xiaopeng,TANG Lixin.An Improves Particle Swarm Optimization Algorithm for Permutation Flowshop Scheduling Problem with Total Flowtime Criterion[J],Lecture Notes in Computer Science,6145,2010,pp.144-151.

[18]Bao Z,Watanabe T.Mixed constrained image filter design using particle swarm optimization[J].Artificial Life and Robotics,Vol15,No.32010,pp.363-368.

[19]Wang Zhongjie ,Zhao Mingming, Hu Miao .Study of the Immune Simulated Annealing Algorithm and Dual-resource Job Shop Order Scheduling[A].Proceedings of the 2013 Chinese Control and Decision Conference (CCDC):2376-2380.

[20]TASGETIREN,M F.,SEVKLI M,LIANG Y C,etal. Particle Swarm Optimization Algorithm for Single Machine Total Weighted Tardiness Problem[C].Proceedings of IEEE Congress on Evolutionary Computation, Washington, D,C., USA: IEEE, 2.2014,pp. 1412-1419.

[21]Carlier J,Pinson.An algorithm for solving the job shop problem,Management Science,36(2),1989,pp.164-176.

[22]E.Taillard."Benchmarks for basic scheduling problems", European Journal of Operational Research,vol.47.1989,pp.67-74.

[23]GAO Shou-wei,LIN Chen,ZHANG Wei-dong.,An Enhanced NEH Method in Solving Permutation Flow Shop Problem[J].Journal of Shanghai Jiaotong University(Science),Vol.E-12,No.1,2007,47-52.

[24]Jun-hua Duan,Min Zhang.A Genetic Algorithm for Permutation Flowshop Scheduling With Total Flowtime Criterion[C].Proceeding of the 2011 Chinese Control and Decision Conference(CCDC),Vol23No.05,2011,pp.1541-1544.