

# OPTIMIZATION AND SIMULATION OF JOB-SHOP SUPPLY CHAIN SCHEDULING IN MANUFACTURING ENTERPRISES BASED ON PARTICLE SWARM OPTIMIZATION

Liao, J.<sup>\*,#</sup> & Lin, C.<sup>\*\*</sup>

\* College of Economics and Business Administration, Chongqing University, Chongqing 400030, China

\*\* Research Center for International Business and Economy, Sichuan International Studies University, Chongqing 400067, China

E-Mail: liaojing@cqu.edu.cn (<sup>#</sup> Corresponding author)

## Abstract

The production scheduling of supply chain is the key to the improvement of production efficiency and resource utilization in manufacturing enterprise. For effective scheduling of job-shop production, this paper puts forward a scheduling optimization method for job-shop supply chain based on particle swarm optimization (PSO), and proves that the PSO is feasible and valid to solve production scheduling problems through example analysis. In addition, a simulation system was established based on the intelligent algorithm and Microsoft SQL Server Platform to solve production scheduling problems. The research shows that the PSO can effectively overcome the nonconvergence problem in production scheduling, and rapidly obtain the optimal solution for job-shop scheduling; the PSO outperforms a common genetic algorithm in the convergence to the optimal solution. The research findings provide a valuable theoretical reference for manufacturing enterprises to solve and optimize production scheduling problems.

(Received, processed and accepted by the Chinese Representative Office.)

**Key Words:** Job-Shop, Supply Chain, Job-Shop Scheduling, Particle Swarm Optimization (PSO), System Simulation

## 1. INTRODUCTION

The production plan of manufacturing enterprises is implemented according to the scheduling of the supply chain. Under the existing production processes and machine conditions, the scheduling performance should be improved comprehensively through rational planning of production activities in light of market or customer demands [1-3]. Production scheduling has been confirmed as the key technique and core issue to the management of manufacturing enterprises [4, 5]. The optimal scheduling plan for job-shop supply chain can effectively enhance customer satisfaction about product delivery, reduce lead time and promote productivity [6, 7]. However, the job-shop scheduling is an NP-hard combinatorial optimization problem in the manufacturing field, which calls for urgent solutions [8-10].

In recent years, many intelligent algorithms, such as simulated annealing algorithm (SAA), genetic algorithm (GA) and ant colony algorithm (ACA), have been applied to production scheduling in manufacturing enterprises, due to the proliferation of computer integrated manufacturing system (CIMS) [11-13]. Nevertheless, there are many defects with these algorithms, including the poor convergence to the optimal solution and the long cycle of problem solving [14, 15]. Compared with these algorithms, the particle swarm optimization (PSO) is a bionic intelligent algorithm that can effectively optimize production scheduling [16, 17]. The PSO can be directly applied to solve continuous optimization problems, because the particle position and velocity in the algorithm are both continuous variables [18, 19].

In light of the above, this paper summarizes the basic theory of the PSO, and designs a PSO algorithm for production scheduling of supply chain in manufacturing enterprise. The

designed algorithm was proved feasible through the simulation on a production scheduling example.

## **2. BASIC THEORY OF THE PSO**

### **2.1 Basic principle of the PSO**

Swarm intelligence emerges in the research of the intelligent phenomenon of colonial organisms in nature: simple individuals can complete complex behaviours through cooperation. In 1995, Kennedy, an American social psychology doctor, proposed the PSO under the inspiration of the movement of organisms in a bird flock. This algorithm compares the random positions in the solution space as the habitats of migrating birds, and predicts the moving direction of the global optimal solution through the interaction between the individuals, thereby finding the best solution. Like the evolutionary algorithm, the PSO allows the individuals to move towards positions with better environmental adaptability. The mathematical model of the PSO is introduced as follows.

In a  $D$ -dimensional search space, an individual is considered as a particle with no volume or mass, and flies at a certain velocity in that space. Let  $N$  be the number of particles in the swarm,  $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  be the coordinates of particle  $i$  in the  $D$ -dimensional search space, and  $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  be the velocity of particle  $i$  in that space. Here, the velocity is defined as the moving distance of a particle in each iteration. Then, the state of particle  $i$  in the  $d$  ( $d = 1 \sim N$ )-dimensional search space can be expressed as:

$$v_{id} = wv_{id} + c_1 rand_1() (P_{id} - x_{id}) + c_2 rand_2() (P_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where  $P_{id}$  is the best-known position of particle  $i$ ;  $P_{gd}$  is the best-known position of the swarm;  $c_1$  and  $c_2$  are acceleration constants;  $w$  is the inertia weight that maintains the inertial movements of the particles;  $rand_1() \sim U(0,1)$  and  $rand_2() \sim U(0,1)$  are two independent, random functions.

In Eq. (1), the movement of particle  $i$  can be divided into three phases. In the first phase, the particle moves under the inertia force and remains in the same state. In the second phase, the particle starts thinking and determines the next movement according to its own information. In the third phase, the particles exchange information and cooperate with each other, and determine their next movements according to the swarm information.

### **2.2 Improved hybrid particle swarm optimization algorithm**

At present, most scholars' optimization of particle swarm optimization is about solving the phenomenon of premature convergence of particles in particle swarm optimization. Because the initial distribution of particles is generated randomly, the initial distribution of particles is unbalanced and the local distribution of the initial population is possible. As a result, particles spend a lot of time in searching and iterating in useless space in the process of iteration search. Even if the crossover and mutation of genetic operators can make the particles jump out of the local optimum situation, it is difficult to ensure that the particles find the optimal solution within a finite iteration because of the randomness of particle diversity.

For the optimization of genetic algorithm, the accuracy of global optimization in genetic algorithm need not be considered, only the optimization speed of the algorithm need to be considered. For improving the genetic algorithm, the fuzzy function is used to increase the adaptive function to the genetic operator of the genetic algorithm, and then the convergence rate of the algorithm speeds up.

Specific ideas as follows, If the average fitness of the parent generation is larger than that of the offspring, it indicates that the result of optimization deviates from the optimal solution, then the crossover and mutation rate of genetic operators are increased. If the average fitness of the parent is less than that of the offspring, it shows that the results of optimization are close to the optimal solution, and the crossover and mutation rates of genetic operators need to be reduced. So a fuzzy function is needed to make the algorithm automatically control the change probability.

The coordinate system  $Z(x, y)$  of the fuzzy model is established,  $x, y \in [-n, n]$ ,  $n$  is a constant. And according to the change of the average value of the offspring particle population, the value should be positive if it needs to be increased and negative if it needs to be reduced.

$$\begin{cases} \Delta cr(t) = Z(x, y) \times c \\ \Delta Pm(t) = Z(x, y) \times p \\ Z = \vartheta x + (1 - \vartheta)y \end{cases} \quad (3)$$

$$\begin{cases} Pm(t) = Pm(t - 1) + \Delta Pm(t) \\ CR(t) = CR(t - 1) + \Delta cr(t) \end{cases} \quad (4)$$

where  $\Delta cr(t)$  represents variation of crossing rate,  $\Delta Pm(t)$  represents variation of mutation rate,  $\vartheta$  is the balance coefficient of  $x$  and  $y$ ,  $c$  is the base value of cross probability,  $p$  is the base value of variation probability.

In addition, in the hybrid algorithm, after calculating the fitness of the particles, the elite set preservation operation is added to transfer the three particles with the highest evaluation value directly to the next generation, so as to prevent the crossover and mutation operation of genetic operators in genetic algorithm from destroying the current better particle genes. So, specific algorithm flow of improved hybrid particle swarm optimization algorithm is shown in Fig. 1.

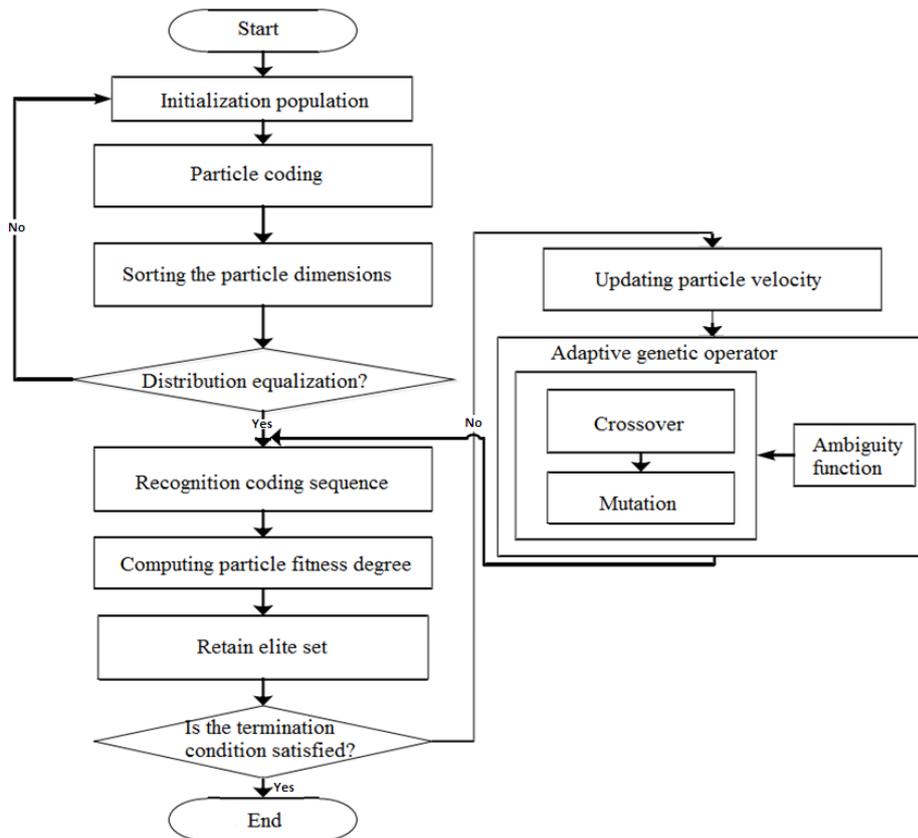


Figure 1: Flow chart of the improved hybrid PSO.

### 3. SCHEDULING OPTIMIZATION OF JOB-SHOP SUPPLY CHAIN

#### 3.1 Production scheduling problem

The production scheduling problem of the supply line in manufacturing enterprises is one of the most classic scheduling problems in the field of CIMS application. The job-shop scheduling usually deals with  $m$  machines and  $n$  jobs that differ in processing sequence. The set of jobs is often denoted as  $P = \{p_1, p_2, \dots, p_n\}$ , where  $p_i$  is the  $i^{\text{th}}$  job ( $i = 1, 2, \dots, n$ ). The set of machines is often denoted as  $M = \{m_1, m_2, \dots, m_m\}$ , where  $m_j$  is the  $j^{\text{th}}$  machine ( $j = 1, 2, \dots, m$ ). The set of processing sequences is often denoted as  $S = \{s_1, s_2, \dots, s_n\}$ , where  $s_i = \{s_{i1}, s_{i2}, \dots, s_{im}\}$  is the processing sequence of job  $p_i$ . Note that  $s_{ik}$  is the serial number of the machine for the  $k^{\text{th}}$  process on the  $i^{\text{th}}$  job ( $k = 1, 2, \dots, m$ ). The production scheduling aims to select the optimal combination between the processing sequences for this problem.

#### 3.2 Coding

There are many coding methods for the job-shop scheduling problem. Here, the process-based expression method is selected as the coding strategy to prevent the omission of the global optimal solution in the coding process and improve the effectiveness and feasibility of the optimization algorithm. The problem in Section 3.1 was subjected to chromosome coding based on processes. Specifically, the jobs were denoted by natural numbers 1, 2, ...,  $n$ , with the number of a job representing the different processes of that job. Each job appeared  $m$  times in the chromosome, and the chromosome was permuted randomly, making it feasible for scheduling. In the decoding process, the chromosome was transformed into an ordered list of processing sequences, and the processes in each phase were implemented at the earliest allowable time according to the decoding result and the processes of each job, forming the final production scheduling plan. Table I describes a three-job, three-machine production scheduling problem, which is taken as an example of job-shop scheduling in the subsequent analysis.

Table I: Production scheduling problem.

Job	Processing sequences and processing time		
$J_1$	$M_3(3)$	$M_1(3)$	$M_2(2)$
$J_2$	$M_1(1)$	$M_3(5)$	$M_2(3)$
$J_3$	$M_2(3)$	$M_1(2)$	$M_3(3)$

In Table I,  $J_i$  and  $M_i$  respectively represent the three jobs and three machines in the production scheduling problem. The three jobs are differentiated by the subscript 1, 2 and 3. During the chromosome coding, 1, 2 and 3 refer to the processing of  $J_1$ ,  $J_2$  and  $J_3$ , respectively. Since each job needs to go through three processes, the same number appears three times in the chromosome. The sequence of that number reflects the process sequence of the corresponding job.

Table II shows the correspondence between jobs, machines and process sequences. In the table, the chromosome is coded as [3 2 2 3 1 3 1 1 2], where the three "1"s stand for the three processes for job  $J_1$ . The first "1" means the first process of job  $J_1$  should be implemented on machine 3; the second and third "1"s mean the second and third processes of  $J_1$  should be implemented on machines 1 and 2, respectively. The machines corresponding to the process sequence of each job can be expressed as [ $M_2 M_1 M_3 M_1 M_3 M_3 M_1 M_2 M_2$ ]. Thus, the process sequence of job  $J_1$  needs to use machines  $M_3 \rightarrow M_1 \rightarrow M_2$ , that of job  $J_2$  needs to use machines  $M_1 \rightarrow M_3 \rightarrow M_2$ , and that of job  $J_3$  needs to use machines  $M_2 \rightarrow M_1 \rightarrow M_3$ . Thus, the job-shop scheduling plan can be plotted as Fig. 2.

Table II: Processing sequences and machines for jobs.

Type	Processing sequences and machines								
Chromosome	3	2	2	3	1	3	1	1	2
Job	$J_3$	$J_2$	$J_2$	$J_3$	$J_1$	$J_3$	$J_1$	$J_1$	$J_2$
Sequence	1	1	2	2	1	3	2	3	3
Machine	$M_2$	$M_1$	$M_3$	$M_1$	$M_3$	$M_3$	$M_1$	$M_2$	$M_2$
Scheduling plan	$J_1: M_3 \rightarrow M_1 \rightarrow M_2; J_2: M_1 \rightarrow M_3 \rightarrow M_2; J_3: M_2 \rightarrow M_1 \rightarrow M_3$								

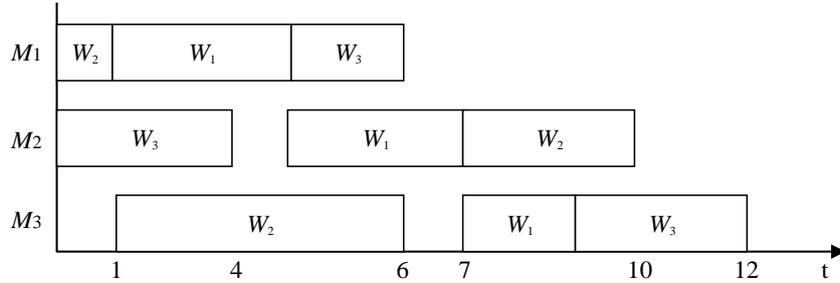


Figure 2: Sketch map of job-shop scheduling plan.

As shown in Fig. 2, the completion times of the three machines  $M_1$ ,  $M_2$  and  $M_3$  are respectively 7, 10 and 12 days. Thus, the earliest completion time is 12 days for all the jobs under the above scheduling plan. This means the above coding method can ensure and effectively utilize the processing sequence of each job, offering a desirable solution to the job-shop scheduling.

### 3.3 Flow of the proposed PSO

When the PSO is adopted to solve the job-shop scheduling problem, the job-shop scheduling plans can be generated by decoding the expression method of the particles. Then, the global and individual optimal particles in the swarm can be determined from the particles in the scheduling plan with the best performance indices. Our PSO for job-shop scheduling contains the following steps: setting the number of particles in the swarm and the maximum number of iterations, initializing the particles, sorting the particles by the position vector, forming an ordered list of processes, generating job-shop scheduling plans by decoding, computing the maximum completion time for processing, and determining the global and individual optimal particles.

Our PSO was applied to solve a five-job, five-machine scheduling problem. The processing time matrix and set of processes of the five jobs can be expressed as:

$$T = \left\{ \begin{array}{l} \{2, 16, 1, 7, 2\} \\ \{9, 4, 1, 1, 4\} \\ \{2, 11, 2, 3, 1\} \\ \{2, 15, 2, 3, 7\} \\ \{1, 5, 10, 5, 1\} \end{array} \right\} \quad (5)$$

$$P = \left\{ \begin{array}{l} \{0, 1, 4, 2, 3\} \\ \{2, 3, 1, 0, 4\} \\ \{1, 0, 2, 4, 3\} \\ \{3, 0, 1, 2, 4\} \\ \{4, 0, 1, 2, 3\} \end{array} \right\} \quad (6)$$

The PSO parameters were set as follows: the number of individuals in the swarm is 20, and the maximum number of iterations is 50 (the algorithm should be terminated after running 50 times). In this way, the worst and best solutions were obtained as 45 and 38, putting the average at 39. This means the mean processing time is 39 days. The convergence of the optimal solution is presented in Fig. 3 below.

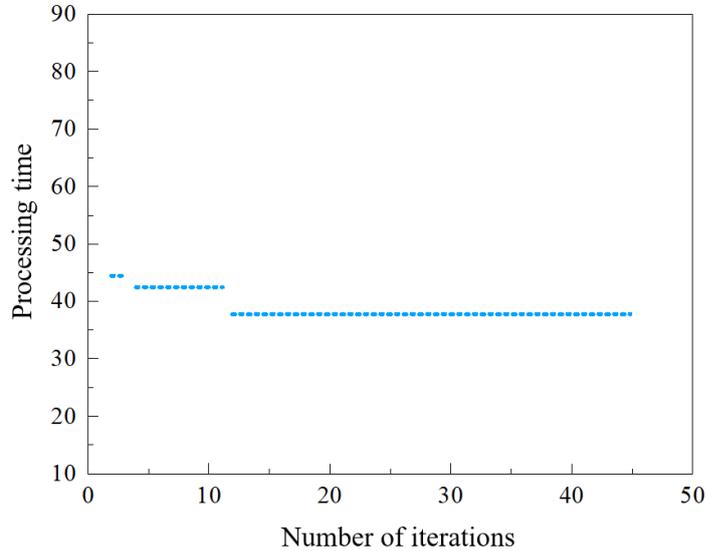


Figure 3: Convergence of the optimal solution.

The same problem was solved again by a common GA. The results in Figs. 4 and 5 show that the best and worst solutions to the job-shop scheduling problem were 50 and 61, respectively, and the mean number of iterations to obtain the optimal solution stood at 18. Comparing the results of our PSO and the common GA, it is clear that the former outperforms in convergence speed, search efficiency and convergence stability.

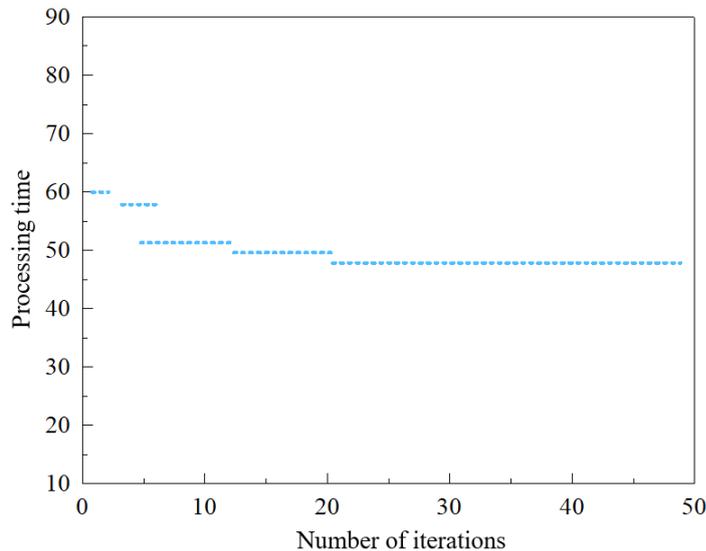


Figure 4: Convergence of the optimal solution (Common GA).

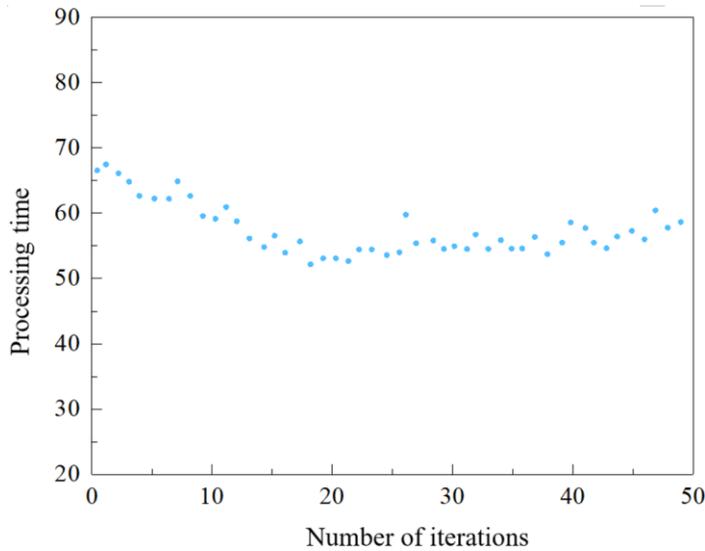


Figure 5: Convergence of the mean value (Common GA).

## 4. SIMULATION OF SUPPLY CHAIN OPTIMIZATION

### 4.1 Construction of simulation model

Any research is inseparable from the abstraction of practical application. The ultimate purpose of the research is to achieve in practical application. In this section, the application environment and preconditions of the improved algorithm are described, and a simulation model suitable for the improved algorithm is constructed.

The simulation model is described as follows: There are  $n$  kinds of workpieces being processed on  $m$  station in a flexible manufacturing system scheduling workshop with multiple automated vehicles. Automated vehicles are responsible for transporting workpieces between multiple stations. Each workpiece has  $n_i$  ( $n_i < m$ ) processes, and the sequence of each workpiece is orderly. Each station is equipped with operation time of different processes, and each station has position coordinates. Workshop architecture diagram is shown in Fig. 6.

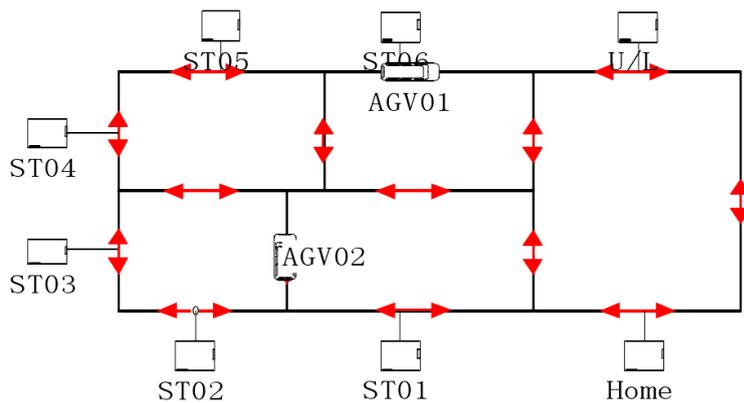


Figure 6: Workshop architecture diagram.

According to the problem description in the practical application scenario and the multi-task scheduling requirements, the related definitions for scheduling evaluation are given.

Definition of particle swarm particle coding: Different processes of the same workpiece is defined by workpiece number,  $Job_i$  represents that there are  $i$  processes, all processes of  $Job_i$  are all defined as  $i$ . For example, there are two processes for  $Job_1$ , two processes for  $Job_2$ , and three processes for  $Job_3$ , then the initial coding order is 1122333.

Definition of task completion time:  $MS$  represents finish time,  $tDT_{ji}^a$  represents the time when No.  $a$  automated vehicle arrives at the designated station to unloads the workpiece of  $O_{ij}$ .  $p_{ij}$  represents the needed time of  $O_{ij}$ .

$$MS = \max\{tDT_{ji}^a + p_{ij}\} \quad a = 1, \dots, z, j = 1, \dots, n, i = 1, \dots, m_j \quad (7)$$

Number of automated vehicles:  $NA$  represents the number of automated vehicles to schedule.

$$NA = \max\{a\} | T = \{T^a\} \wedge T^a = \{T_{ji}^a | A^a \text{ is assigned to Task } T_{ji}\} \quad (8)$$

Definition of multi-objective evaluation: In the posterior method of multi-objective evaluation, Pareto is applied as a more optimized method for multi-objective evaluation. Pareto method formula is expressed as:  $f(x) = \sum_{\beta=1}^{\beta=L} \vartheta_{\beta} \omega_{\beta} f_{\beta}(x)$ , where  $\omega_{\beta}$  and  $\vartheta_{\beta}$  is a pair of weight parameters. In our model, number of automated vehicles is set to 1/3, the weight of task completion time is set to 2/3. The fitness calculation function is:

$$f(x) = \frac{2}{3}(MS) + \frac{1}{3}(NA) \quad (9)$$

### 4.2 Simulation results and analysis

Firstly, the relevant data used in the initialization of simulation are given. Table III is the station location information table, which records the absolute distance of ST from each workstation to L/U of loading and unloading workstation.

Table III: Station location information.

	L/U	ST <sub>1</sub>	ST <sub>2</sub>	ST <sub>3</sub>	ST <sub>4</sub>	ST <sub>5</sub>	ST <sub>6</sub>
L/U	0	4	6	9	11	16	18
ST <sub>1</sub>	14	0	1	4	6	11	13
ST <sub>2</sub>	12	17	0	2	4	9	11
ST <sub>3</sub>	9	14	16	0	1	6	8
ST <sub>4</sub>	7	12	14	17	0	4	6
ST <sub>5</sub>	2	7	9	12	14	0	1
ST <sub>6</sub>	1	5	7	10	12	17	0

Table IV is process and station corresponding table (see on the next page). The discrete event simulation result is shown in Fig. 7. The same colour block in Fig. 7 represents the same workpiece, the horizontal ordinate corresponds to the processing time and the ordinate corresponds to the processing station. Arrow represents automated vehicle, and the folded line formed by the same colour arrow records the task scheduling route of automated vehicle.

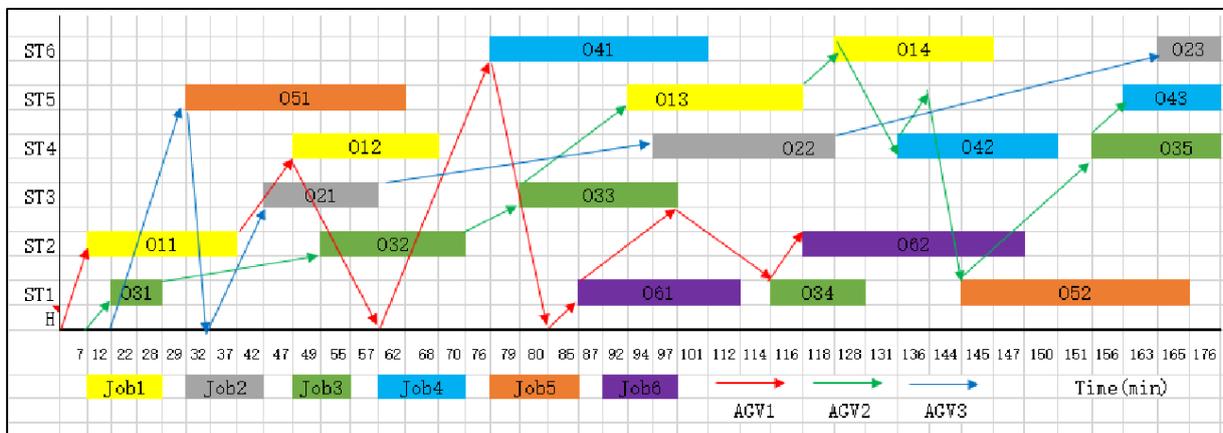


Figure 7: The discrete event simulation result.

Table IV: Process and station comparison.

Workpiece	Process	Station	Time (Minute)
1	1	ST <sub>2</sub>	28
1	2	ST <sub>4</sub>	20
1	3	ST <sub>5</sub>	18
1	4	ST <sub>6</sub>	22
2	1	ST <sub>3</sub>	16
2	2	ST <sub>4</sub>	12
2	3	ST <sub>6</sub>	14
3	1	ST <sub>1</sub>	20
3	2	ST <sub>2</sub>	17
3	3	ST <sub>3</sub>	15
3	4	ST <sub>1</sub>	25
3	5	ST <sub>4</sub>	20
4	1	ST <sub>6</sub>	17
4	2	ST <sub>4</sub>	18
4	3	ST <sub>5</sub>	21
5	1	ST <sub>5</sub>	15
5	2	ST <sub>1</sub>	16
6	1	ST <sub>1</sub>	21
6	2	ST <sub>2</sub>	17

The simulation results show that the improved hybrid particle swarm optimization algorithm can find a better solution with better fitness, and the convergence speed is obviously faster, task completion time is shorter, and the number of automated vehicle scheduling is less.

## **5. CONCLUSIONS**

Inspired by the PSO theory, this paper puts forward a PSO-based scheduling optimization method for job-shop supply chain, and applies the method in a simulation of production scheduling problems. The main conclusions are as follows.

The proposed PSO can effectively overcome the nonconvergence problem in production scheduling. The algorithm contains the following steps: setting the number of particles in the swarm and the maximum number of iterations, initializing the particles, sorting the particles by the position vector, forming an ordered list of processes, generating job-shop scheduling plans by decoding, computing the maximum completion time for processing, and determining the global and individual optimal particles.

In job-shop scheduling, the proposed PSO outperforms the common GA in search efficiency and convergence effect, especially in the convergence speed to the optimal solution, and convergence stability. The authors developed a simulation system for the scheduling optimization of job-shop supply chain based on the Microsoft SQL Server Platform of Windows XP, using ADO in C++.

## **REFERENCES**

- [1] Chen, A. L. (2014). Partner selection of agricultural product supply chain based on discrete particle swarm optimization algorithm, *Applied Mechanics and Materials*, Vol. 670-671, 1633-1636, doi:[10.4028/www.scientific.net/amm.670-671.1633](https://doi.org/10.4028/www.scientific.net/amm.670-671.1633)
- [2] Wang, S.; Mao, C. (2018). Evaluation of regional manufacturing quality competitiveness based on analytic network, *Journal Européen des Systèmes Automatisés*, Vol. 51, No. 1-3, 109-124, doi:[10.3166/JESA.51.109-124](https://doi.org/10.3166/JESA.51.109-124)

- [3] Che, Z. H. (2010). Using fuzzy analytic hierarchy process and particle swarm optimisation for balanced and defective supply chain problems considering WEEE/RoHS directives, *International Journal of Production Research*, Vol. 48, No. 11, 3355-3381, doi:[10.1080/00207540802702080](https://doi.org/10.1080/00207540802702080)
- [4] Kannan, G.; Noorul-Haq, A.; Devika, M. (2009). Analysis of closed loop supply chain using genetic algorithm and particle swarm optimisation, *International Journal of Production Research*, Vol. 47, No. 5, 1175-1200, doi:[10.1080/00207540701543585](https://doi.org/10.1080/00207540701543585)
- [5] Soleimani, H.; Kannan, G. (2015). A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks, *Applied Mathematical Modelling*, Vol. 39, No. 14, 3990-4012, doi:[10.1016/j.apm.2014.12.016](https://doi.org/10.1016/j.apm.2014.12.016)
- [6] Sinha, A. K.; Aditya, H. K.; Tiwari, M. K.; Chan, F. T. S. (2011). Agent oriented petroleum supply chain coordination: Co-evolutionary particle swarm optimization based approach, *Expert Systems with Applications*, Vol. 38, No. 5, 6132-6145, doi:[10.1016/j.eswa.2010.11.004](https://doi.org/10.1016/j.eswa.2010.11.004)
- [7] Sinha, A. K.; Aditya, H. K.; Tiwari, M. K.; Chan, F. T. S. (2010). Multi-agent based petroleum supply chain coordination: A co-evolutionary particle swarm optimization approach, *Proceedings of the World Congress on Nature & Biologically Inspired Computing*, 1349-1354, doi:[10.1109/nabic.2009.5393741](https://doi.org/10.1109/nabic.2009.5393741)
- [8] Sarmiento, A. T.; Rabelo, L.; Jones, A. (2009). Using particle swarm optimization with a policy optimization approach to stabilize the supply chain, *Proceedings of the 2009 Winter Simulation Conference*, 851-862
- [9] PrasannaVenkatesan, S.; Kumanan, S. (2012). Multi-objective supply chain sourcing strategy design under risk using PSO and simulation, *The International Journal of Advanced Manufacturing Technology*, Vol. 61, No. 1-4, 325-337, doi:[10.1007/s00170-011-3710-y](https://doi.org/10.1007/s00170-011-3710-y)
- [10] Kuo, R. J.; Huang, C. C. (2009). Application of particle swarm optimization algorithm for solving bi-level linear programming problem, *Computers & Mathematics with Applications*, Vol. 58, No. 4, 678-685, doi:[10.1016/j.camwa.2009.02.028](https://doi.org/10.1016/j.camwa.2009.02.028)
- [11] Mohanty, C. P.; Mahapatra, S. S.; Singh, M. R. (2016). A particle swarm approach for multi-objective optimization of electrical discharge machining process, *Journal of Intelligent Manufacturing*, Vol. 27, No. 6, 1171-1190, doi:[10.1007/s10845-014-0942-3](https://doi.org/10.1007/s10845-014-0942-3)
- [12] Rabelo, L.; Sarmiento, A. T.; Jones, A. (2011). Stability of the supply chain using system dynamics simulation and the accumulated deviations from equilibrium, *Modelling and Simulation in Engineering*, Vol. 2011, Paper 603632, 10 pages, doi:[10.1155/2011/603632](https://doi.org/10.1155/2011/603632)
- [13] Gopi, A. P.; Narayana, V. L.; Kumar, N. A. (2018). Dynamic load balancing for client server assignment in distributed system using genetic algorithm, *Ingénierie des Systèmes d'Information*, Vol. 23, No. 6, 87-98. doi:[10.3166/ISI.23.6.87-98](https://doi.org/10.3166/ISI.23.6.87-98)
- [14] Che, Z. H.; Tzu-An, C.; Kuo, Y. C.; Zhihua, C. (2014). Hybrid algorithms for fuzzy reverse supply chain network design, *The Scientific World Journal*, Vol. 2014, Paper 497109, 16 pages, doi:[10.1155/2014/497109](https://doi.org/10.1155/2014/497109)
- [15] Chan, F. T. S.; Chung, S. H. (2007). A multi-criterion genetic algorithm for order distribution in a demand driven supply chain, *International Journal of Computer Integrated Manufacturing*, Vol. 17, No. 4, 339-351, doi:[10.1080/09511920310001617022](https://doi.org/10.1080/09511920310001617022)
- [16] D'Amours, S.; Frayret, J.-M.; Rousseau, A.; Harvey, S.; Plamondon, P.; Forget, P. (2006). Agent-based supply-chain planning in the forest products industry, *Proceedings of the International Conference on Information Technology for Balanced Automation Systems*, 17-26, doi:[10.1007/978-0-387-36594-7\\_3](https://doi.org/10.1007/978-0-387-36594-7_3)
- [17] Fu, H.; Xu, J.; Zhang, H.; Zhang, M.; Xu, X. (2018). A novel video target tracking method based on lie group manifold, *Traitement du Signal*, Vol. 35, No. 3-4, 331-340, doi:[10.3166/TS.35.331-340](https://doi.org/10.3166/TS.35.331-340)
- [18] Schliephake, K.; Stevens, G.; Clay, S. (2009). Making resources work more efficiently – the importance of supply chain partnerships, *Journal of Cleaner Production*, Vol. 17, No. 14, 1257-1263, doi:[10.1016/j.jclepro.2009.03.020](https://doi.org/10.1016/j.jclepro.2009.03.020)
- [19] Govindu, R.; Chinnam, R. B. (2007). MASCF: A generic process-centered methodological framework for analysis and design of multi-agent supply chain systems, *Computers & Industrial Engineering*, Vol. 53, No. 4, 584-609, doi:[10.1016/j.cie.2007.06.003](https://doi.org/10.1016/j.cie.2007.06.003)