

# FLEXIBLE JOB-SHOP SCHEDULING PROBLEM BASED ON HYBRID ACO ALGORITHM

Wu, J.<sup>\*</sup>; Wu, G. D.<sup>\*\*</sup> & Wang, J. J.<sup>\*\*\*,#</sup>

<sup>\*</sup> School of Management, Hangzhou Dianzi University, Hangzhou 310018, China

<sup>\*\*</sup> School of Tourism and Urban Management, Jiangxi University of Finance & Economics, Nanchang 330013, China

<sup>\*\*\*</sup> College of Information & Business, Zhengzhou University of Technology, Zhengzhou 451191, China

Email: wangjun718100@126.com (<sup>#</sup> Corresponding author)

## Abstract

For an enterprise to survive the fierce market competition, efficient production scheduling is a must as it improves economic efficiency and reduces cost. As an important branch of production scheduling, the flexible job-shop scheduling problem (FJSP) is a mixed blessing. It accurately reflects the characteristics of the actual production, but adds to the difficulties in problem solving. With the ant colony algorithm as the basic optimization method, this paper proposes the hybrid ant colony algorithm based on the 3D disjunctive graph model by combining the elitist ant system, max-min ant system and the staged parameter control mechanism, optimizes the FJSP problem to minimize the longest completion time, the early/delay penalty cost, the average idle time of the machine, and the production cost, and verifies the effectiveness of the model and algorithm by an example.

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

**Key Words:** Flexible Job-Shop Scheduling Problem (FJSP), Multi-Objective Optimization, Hybrid Ant Colony Algorithm

## 1. INTRODUCTION

Scheduling is a common problem in all areas of economics, from computer engineering to manufacturing. In most cases, it is a complex optimization problem too difficult to be resolved effectively. The most difficult combinatorial optimization problem is job-shop scheduling problems (JSP), which is a branch of production scheduling. It is well documented that JSP is an NP-hard problem [1]. It is impossible to get the optimal result through the existing algorithms even if the problem is only a medium-sized one. Recent years have seen a growing concern over the flexible job-shop scheduling problem (FJSP) [2-4], which is an extension of the JSP problem. FJSP mainly contains two kinds of problems, both of which aim at minimizing the predefined targets. One is to allocate a machine to each operation; the other is to arrange the order of the operations on each machine.

Brucker and Schlie were the first to raise the problem, and they developed a polynomial algorithm to solve it [5]. Brandimarte was the first scholar to apply the decomposition method to solve FJSP. He used some of the existing scheduling rules and tabu search heuristic algorithm to solve the routing sub-problem of FJSP [6]. Tung et al. put forward a similar scheduling method for flexible manufacturing systems [7]. Kacem et al. proposed a genetic algorithm controlled by the allocation model and generated by the localization method. Through comprehensive consideration, the method can realize the allocation and scheduling at the same time [8]. Viewing reassignment and re-scheduling as two different types of actions, Hurink et al. came up with the tabu search heuristic algorithm [9]. Dauzere-Peres and Paulli proposed a community structure and integration method without differentiating redistribution from re-scheduling [10]. Mastrolilli and Gambardella improved the tabu search technology of Dauzere-Peres and presented two community functions [11].

It should be pointed out that most of the previous research only pursues a single optimization target [12-16], and the existing models lack versatility due to the lack of constraints on actual production [17-20]. Therefore, this study attempts to solve the multi-objective FJSP problem through the ant colony algorithm. Specifically, the hybrid ant colony algorithm is designed based on the 3D disjunctive graph model to optimize four FJSP targets, and the model and algorithm are verified with an example.

## 2. INTRODUCTION OF FJSP

### 2.1 Problem description

The FJSP problem is about the processing of  $n$  workpieces with a total of  $m$  machines in the system. Each workpiece requires one or more operations, and each operation can work on different machines (following the preset processing order). Plus, the processing time depends on the performance of specific machines. In essence, the problem is equivalent to the optimization of the overall performance indices of the system by assigning the operations to machines, setting the start time of the operation, and determining the processing order [21-22].

The following assumptions are made before modelling: the machines are independent of each other; the time for different operations to move on different machines is negligible; a machine can only handle one workpiece or one operation at a given time; the different operations for the same workpiece should follow a strict order to meet the corresponding time constraints, but the constraints do not apply to different workpieces.

### 2.2 Mathematical modelling

During the processing of parts and components, all departments of an enterprise naturally desire that the job shop scheduling decisions are in line with their own interests. In order to balance the interests of multiple parties, this paper establishes the multi-objective optimization model to minimize the longest completion time, the early/delay penalty cost, the average idle time of the machine, and the production cost. The specific steps are as follows.

Objective functions:

$$\min f_1 = \max\left\{\sum_{i=1}^n C_i\right\} \quad (1)$$

$$\min f_2 = \min\left\{\sum_{i=1}^n \chi_i * \max(0, E_i - C_i) + \mu_i * \max(T_i - C_i, 0)\right\} \quad (2)$$

$$\min f_3 = \min\left\{\sum_{i=1}^m I_i / m\right\} (I_i = C_{\max} - \sum_{j=1}^{m_i} C_{ij}, C_{\max} = \max\{C_1, \dots, C_n\}) \quad (3)$$

$$\min f_4 = \min\left\{\sum_{i=1}^m v_i WF_i + \sum_{i=1}^m v'_i (W_i - WF_i)\right\} \quad (4)$$

Constraint conditions:

$$E_{ijk} - E_{i(j-1)k} > c_{ijk} \quad (5)$$

where  $1 < j < j_i$ ,  $Y_{ijk} = Y_{i(j-1)m} = 1$ .

$$E_{egk} - E_{ijk} > c_{egk} \quad (6)$$

where  $Y_{ijk} = Y_{ejk} = 1$ ,  $X_{ijegk} = 1$ .

$$E_{ijk} > c_{egk} \quad (7)$$

where  $Y_{ijl} = 1$ .

The objective functions  $f_1, f_2, f_3$  and  $f_4$  stand for the minimum longest completion time, the minimum early/delay penalty cost, the minimum average idle time of the machine, and the minimum production cost, respectively. Constraint condition (6) means the  $j^{\text{th}}$  operation of workpiece  $J_i$  should not start before the completion of the  $j-1^{\text{th}}$  operation; constraint condition (7) means machine  $k$  should not handle two different workpieces/operations simultaneously at any given time; constraint condition (8) means the processing time of any operation must be shorter than its completion time.

In the above formulas:  $C_i$  is the completion time of workpiece  $J_i$ ;  $\chi_i$  is the unit time penalty coefficient for early completion of workpiece  $J_i$ ;  $\mu_i$  is the unit time penalty coefficient for delayed completion of workpiece  $J_i$ ;  $E_i$  is the earliest delivery time of workpiece  $J_i$ ;  $T_i$  is the latest delivery time of workpiece  $J_i$ ;  $I_i$  is the idle time of machine  $M_i$ ;  $C_{max}$  is the maximum completion time for all workpieces;  $C_{ij}$  is the completion time for the  $j^{\text{th}}$  workpiece on machine  $M_i$ ;  $m_i$  is the number of workpieces processed on machine  $M_i$ ;  $WF_i$  is the sum of all processing times on machine  $M_i$ ;  $W_i$  is the completion time of the last workpiece processed on machine  $M_i$ ;  $v_i, v'_i$  are the dynamic and static rates of machine  $M_i$ , respectively;  $n$  is the number of workpieces;  $m$  is the number of machines;  $J_i$  is the  $i^{\text{th}}$  workpiece;  $j_i$  is the number of operations contained in workpiece  $J_i$ ;  $C_{ijk}$  is the operation time for machine  $k$  to handle the  $j^{\text{th}}$  operation of workpiece  $J_i$ ;  $E_{ijk}$  is the processing time for machine  $k$  to handle the  $j^{\text{th}}$  operation of workpiece  $J_i$ ;  $MF_k$  is the sum of all processing times for all workpieces on machine  $k$ ;  $MF$  is the sum of all processing times for all workpieces.

$$X_{ijegk} = \begin{cases} 1, & \text{Handle the } j^{\text{th}} \text{ operation of workpiece } i \text{ and the } g^{\text{th}} \text{ operation of workpiece } e \text{ on the} \\ & \text{machine. If the processing time of workpiece } j \text{ is earlier than } g, X_{ijegk} = 1. \\ 0, & \text{Otherwise, } X_{ijegk} = 0. \end{cases}$$

$$Y_{ijk} = \begin{cases} 1, & \text{If the } j^{\text{th}} \text{ operation of workpiece } i \text{ is handled on the machine, } Y_{ijk} = 1. \\ 0, & \text{Otherwise, } Y_{ijk} = 0. \end{cases}$$

### 3. ALGORITHM DESIGN

Ant colony optimization (ACO) is an evolutionary algorithm derived from the path selection behaviour of ants searching for food. Despite the many good features revealed by existing research, the algorithm often converges to the local optimal solution during the solution of large-scale problems. With the aim of overcoming this shortcoming, this paper sets up the following mixed ACO algorithm, hoping to obtain a better performing solution to the FJSP problem.

#### 3.1 3D disjunctive graph modelling

Disjunctive graph modelling is essential to the solution of optimization problems by the ACO. In view of the lack of information and poor effect of 2D disjunctive graph-based models for the FJSP problem, this paper proposes a method for modelling based on the 3D disjunctive graph. Fig. 1 illustrates the details of the model.

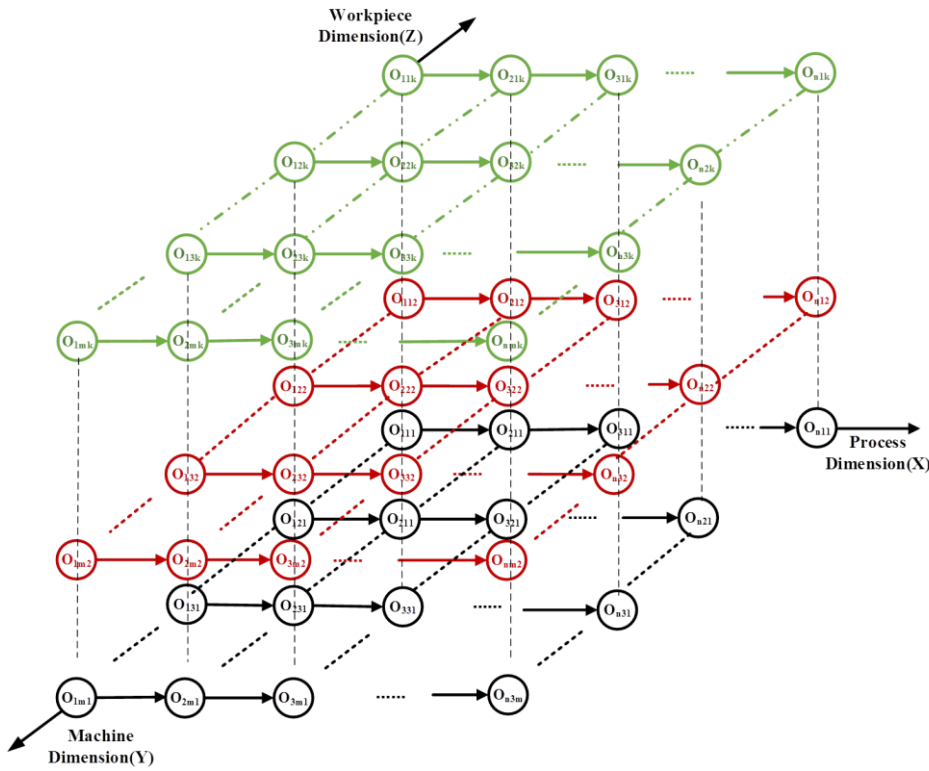


Figure 1: 3D disjunctive graph-based model for FJSP.

### 3.2 Algorithm flowchart

Fig. 2 shows the flowchart of the algorithm.

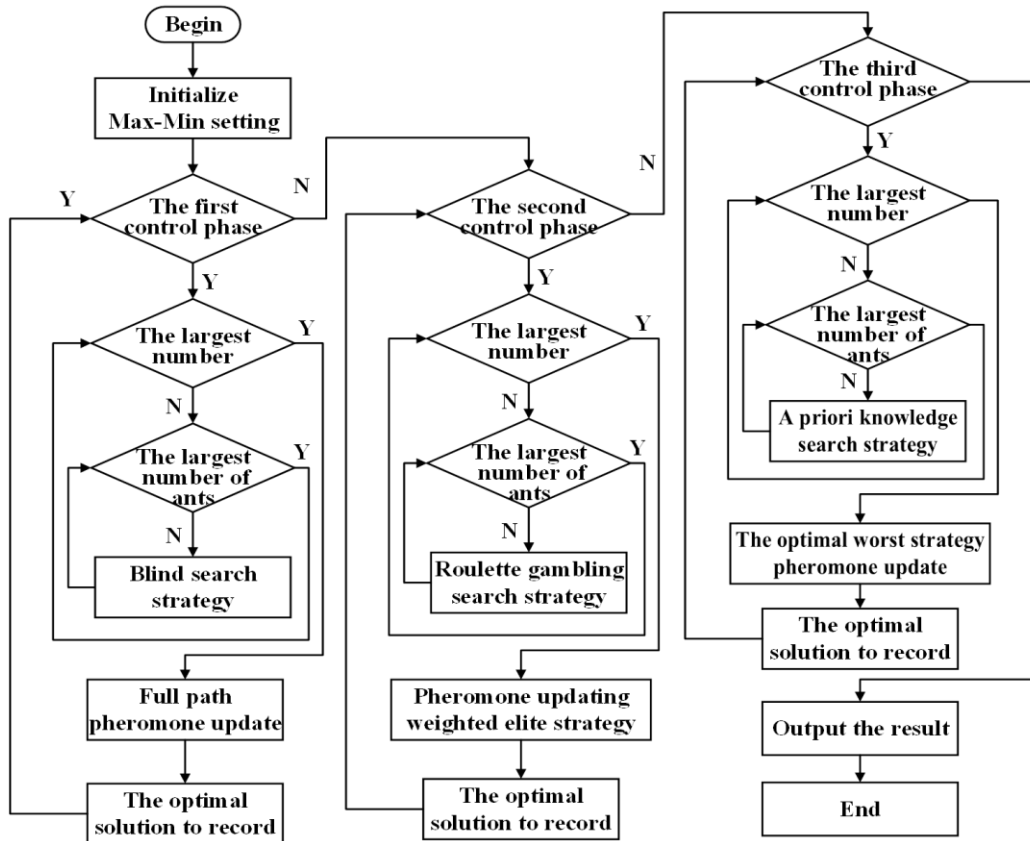


Figure 2: Flowchart of the algorithm.

The steps are as follows:

**Step 1:** Set the parameters of the algorithm, including the number of cycles  $N$ , the number of ants in the colony  $L$ , the control parameters in the stage parameter control mechanism  $\delta_1, \delta_2$ , the volatilization coefficients of pheromone that enable information exchange between individual ants  $\rho_1, \rho_2, \rho_3$  the node of ant  $l$ , the basic quantities of pheromone  $Q_1, Q_2, Q_3$ , the ant number  $m_1, \dots, m_L$ , as well as the parameters of the maximum and minimum ant colony mechanism *Max-Min*:  $(\varepsilon_{min1}, \varepsilon_{max1}), (\varepsilon_{min2}, \varepsilon_{max2}), (\varepsilon_{min3}, \varepsilon_{max3}), \Delta\tau_{ij} = 0$  and  $t = 1$ .

**Step 2:** Random search phase (cycle 1 to cycle  $N - \rho_1 N$ )

(1) Set the parameters of the *Max-Min* mechanism  $\varepsilon_{min1}, \varepsilon_{max1}$ ;

(2) Conduct random search with artificial ants: When an ant arrives at a new node, judge if the ant can support plug-in search; if yes, proceed with the search; otherwise, let the ant  $m_w$  find the next node in the candidate strategy set  $D_v^w$  following the random seeking strategy. Place the node into the result set  $R_w$ , and let the ant move forward by a node. Form a new candidate strategy set according to the tracing rules of the 3D disjunctive graph;

(3) Pheromone update: Update the pheromone of all ants based on the solution of the current cycle and the basic pheromone strategy.

**Step 3:** Roulette search phase (cycle  $N - \delta_1 N$  to cycle  $\delta_1 N - \delta_2 N$ )

(1) Set the parameters of the *Max-Min* mechanism  $\varepsilon_{min2}, \varepsilon_{max2}$ ;

(2) Conduct roulette search with artificial ants: When an ant arrives at a new node, judge if the ant can support plug-in search; if yes, proceed with the search; otherwise, let the ant  $m_w$  find the next node in the candidate strategy set  $D_v^w$  following the roulette strategy. Place the node into the result set  $R_w$ , and let the ant move forward by a node. Form a new candidate strategy set according to the tracing rules of the 3D disjunctive graph;

(3) Pheromone update: Update the pheromone of all ants based on the solution of the current cycle and the weighted elitist strategy.

**Step 4:** Prior knowledge search phase (cycle  $\delta_1 N - \delta_2 N$  to cycle  $N$ )

(1) Set the parameters of the *Max-Min* mechanism  $\varepsilon_{min3}, \varepsilon_{max3}$ ;

(2) Conduct prior knowledge search with artificial ants: When an ant arrives at a new node, let the ant  $m_w$  find the next node in the candidate strategy set  $D_v^w$  following the idea of prior knowledge. Place the node into the result set  $R_w$ , and let the ant move forward by a node. Form a new candidate strategy set according to the tracing rules in Fig. 1;

(3) Pheromone update: Update the pheromone of all ants based on the solution of the current cycle and the best-worst mechanism.

**Step 5:** Terminate the calculation and output the results.

## 4. CASE STUDY

### 4.1 Example selection

An  $8 \times 8$  FJSP problem is chosen. See Table I for the processing time of each operation on each machine.

Table I: The 8×8 FJSP problem with 27 operations.

Workpieces	Operation	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$
$J_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$	10	9
	$O_{1,2}$	10		5	8	3	9	9	6
	$O_{1,3}$		10		5	6	2	4	2
$J_2$	$O_{2,1}$	5	7	3	9	8		9	
	$O_{2,2}$		8	5	2	6	7	10	9
	$O_{2,3}$		10		5	6	4	1	7
	$O_{2,4}$	10	8	9	6	4	7		
$J_3$	$O_{3,1}$	10			7	6	5	2	4
	$O_{3,2}$		10	6	4	8	9	10	
	$O_{3,3}$	1	4	5	6		10		7
$J_4$	$O_{4,1}$	3	1	6	5	9	7	8	4
	$O_{4,2}$	12	11	7	8	10	5	6	9
	$O_{4,3}$	4	6	2	10	3	9	5	7
$J_5$	$O_{5,1}$	3	6	7	8	9		10	
	$O_{5,2}$	10		7	4	9	8	6	
	$O_{5,3}$		9	8	7	4	2	7	
	$O_{5,4}$	11	9		6	7	5	3	6
$J_6$	$O_{6,1}$	6	7	1	4	6	9		10
	$O_{6,2}$	11		9	9	9	7	6	4
	$O_{6,3}$	10	5	9	10	11		10	
$J_7$	$O_{7,1}$	5	4	2	6	7		10	
	$O_{7,2}$		9		9	11	9	10	5
	$O_{7,3}$		8	9	3	8	6		10
$J_8$	$O_{8,1}$	2	8	5	9		4		10
	$O_{8,2}$	7	4	7	8	9		10	
	$O_{8,3}$	9	9		8	5	6	7	1
	$O_{8,4}$	9		3	7	1	5	8	

Table II and Table III list the specific parameters.

Table II: The rate of each machine (RMB yuan/unit time).

Machine	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$	$M_7$	$M_8$
Dynamic rate	5	1	3	4	10	8	6	5
Static rate	1	0.2	1	1	2	1	2	0.5

Table III: The weight of the optimization index.

Optimization index	$\min f_1$	$\min f_2$	$\min f_3$	$\min f_4$
Weight	$\lambda_1 = 0.2$	$\lambda_2 = 0.3$	$\lambda_3 = 0.2$	$\lambda_4 = 0.3$
Comprehensive index	$\min f = \sum_{i=1}^4 \theta_i \cdot \lambda_i \cdot \min f_i$ ( $\theta_i$ : Dimension coefficient)			

### 4.2 Example results

Run the algorithm in 100 cycles for 10 times to get the optimal solution (Table IV). The trend of the optimal solution in the cycles is illustrated in Fig. 3. Fig. 4 is the Gantt chart.

Table IV: The optimal solution.

$\min f$	$\min f_1$	$\min f_2$	$\min f_3$	$\min f_4$
272.37	40	10.7	24.38	865.4

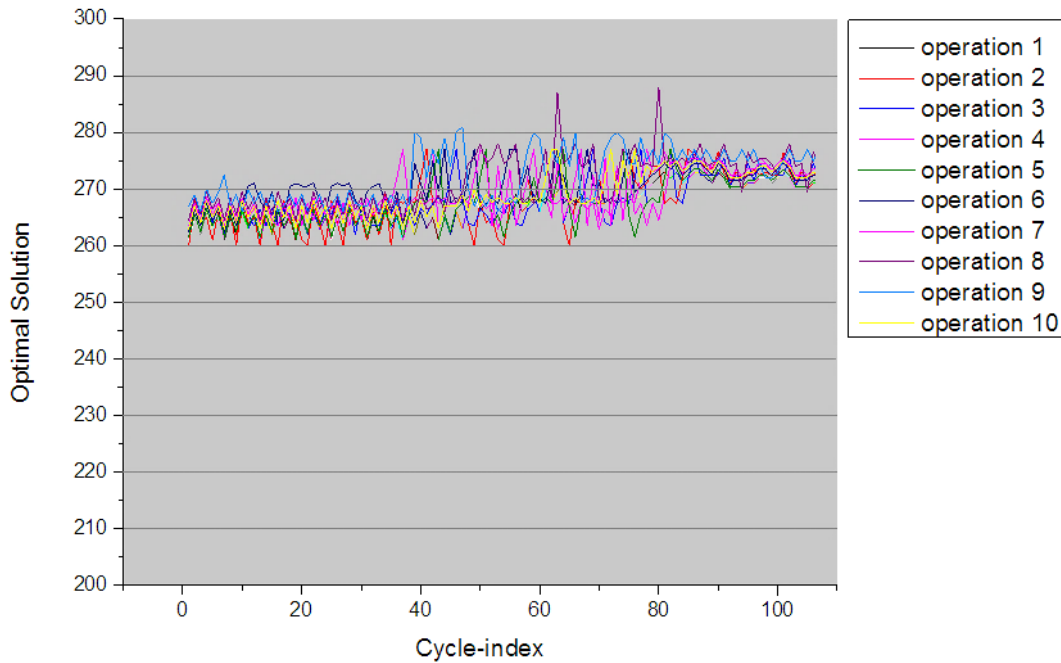


Figure 3: The trend of optimal solution in the cycles.

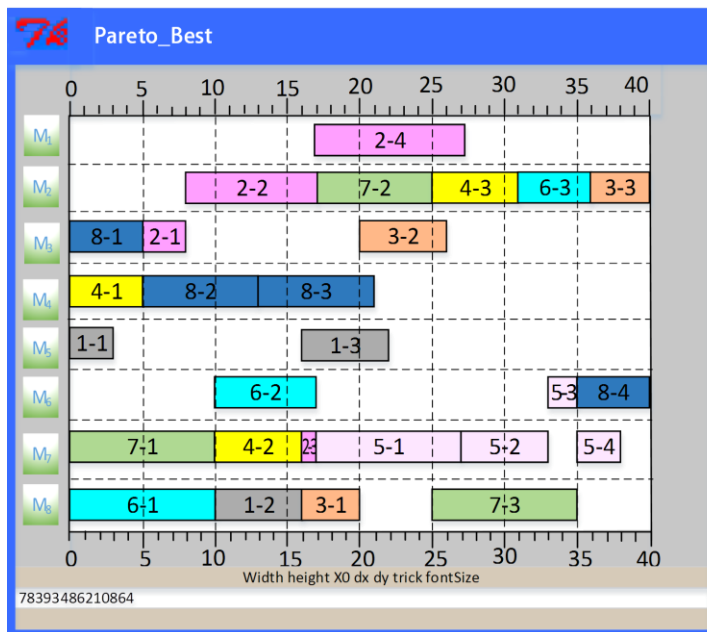


Figure 4: Gantt chart of Pareto.

As stated above, the minimum longest completion time, the minimum early/delay penalty cost, the minimum average idle time of the machine, and the minimum production cost of the

FJSP problem are 40.00, 10.70, 24.38 and 865.40, respectively. Based on the four indices, the comprehensive index is calculated as 272.37. The Gantt chart shows the processing order of the operations of different workpieces on each machine.

## **5. CONCLUSIONS**

The FJSP problem accurately illustrates the actual production scheduling of manufacturing systems, and efficient and rational production scheduling is the key to improving the market competitiveness of enterprises. In light of this, this paper focuses on the FJSP problem, explains how to solve the problem from the aspects of problem description, modelling and algorithm design, and verifies the effectiveness of the proposed algorithm with an example.

## **ACKNOWLEDGEMENT**

This work was supported by Zhejiang Provincial Natural Science Foundation of China research program (Grant No. LQ15G020006) and Philosophy and Social Science Plan of Zhejiang Province (Grant No. [2015] NO.7).

## **REFERENCES**

- [1] Garey, M. R.; Johnson, D. S.; Sethi, R. (1976). The complexity of flowshop and jobshop scheduling, *Mathematics of Operations Research*, Vol. 1, No. 2, 117-129, doi:[10.1287/moor.1.2.117](https://doi.org/10.1287/moor.1.2.117)
- [2] Chan, F. T. S.; Wong, T. C.; Chan, L. Y. (2006). Flexible job-shop scheduling problem under resource constraints, *International Journal of Production Research*, Vol. 44, No. 11, 2071-2089, doi:[10.1080/00207540500386012](https://doi.org/10.1080/00207540500386012)
- [3] Zhang, C.; Rao, Y.; Li, P.; Shao, X. (2007). Bilevel genetic algorithm for the flexible job-shop scheduling problem, *Chinese Journal of Mechanical Engineering*, Vol. 43, No. 4, 119-124
- [4] Zhang, G.; Gao, L.; Li, P.; Zhang, C. (2009). Improved genetic algorithm for the flexible job-shop scheduling problem, *Chinese Journal of Mechanical Engineering*, Vol. 45, No. 7, 145-151
- [5] Brucker, P.; Schlie, R. (1990). Job-shop scheduling with multi-purpose machines, *Computing*, Vol. 45, No. 4, 369-375, doi:[10.1007/bf02238804](https://doi.org/10.1007/bf02238804)
- [6] Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search, *Annals of Operations Research*, Vol. 41, No. 3, 157-183, doi:[10.1007/bf02023073](https://doi.org/10.1007/bf02023073)
- [7] Tung, L.-F.; Lin, L.; Nagi, R. (1999). Multiple-objective scheduling for the hierarchical control of flexible manufacturing systems, *International Journal of Flexible Manufacturing Systems*, Vol. 11, No. 4, 379-409, doi:[10.1023/a:1008183330452](https://doi.org/10.1023/a:1008183330452)
- [8] Kacem, I.; Hammadi, S.; Borne, P. (2002). Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic, *Mathematics and Computers in Simulation*, Vol. 60, No. 3-5, 245-276, doi:[10.1016/s0378-4754\(02\)00019-8](https://doi.org/10.1016/s0378-4754(02)00019-8)
- [9] Hurink, J.; Jurisch, B.; Thole, M. (1994). Tabu search for the job-shop scheduling problem with multi-purpose machines, *OR Spectrum*, Vol. 15, No. 4, 205-215, doi:[10.1007/bf01719451](https://doi.org/10.1007/bf01719451)
- [10] Dauzère-Pérès, S.; Paulli, J. (1997). An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search, *Annals of Operations Research*, Vol. 70, 281-306, doi:[10.1023/a:1018930406487](https://doi.org/10.1023/a:1018930406487)
- [11] Mastrolilli, M.; Gambardella, L. M. (2000). Effective neighbourhood functions for the flexible job shop problem, *Journal of Scheduling*, Vol. 3, No. 1, 3-20, doi:[10.1002/\(sici\)1099-1425\(200001/02\)3:1<3::aid-jos32>3.0.co;2-y](https://doi.org/10.1002/(sici)1099-1425(200001/02)3:1<3::aid-jos32>3.0.co;2-y)
- [12] Amiri, M.; Zandieh, M.; Yazdani, M.; Bagheri, A. (2010). A variable neighbourhood search algorithm for the flexible job-shop scheduling problem, *International Journal of Production Research*, Vol. 48, No. 19, 5671-5689, doi:[10.1080/00207540903055743](https://doi.org/10.1080/00207540903055743)



- [13] Moslehi, G.; Mahnam, M. (2011). A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, *International Journal of Production Economics*, Vol. 129, No. 1, 14-22, doi:[10.1016/j.ijpe.2010.08.004](https://doi.org/10.1016/j.ijpe.2010.08.004)
- [14] Rajkumar, M.; Asokan, P.; Anilkumar, N.; Page, T. (2011). A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints, *International Journal of Production Research*, Vol. 49, No. 8, 2409-2423, doi:[10.1080/00207541003709544](https://doi.org/10.1080/00207541003709544)
- [15] Wang, L.; Wang, S.; Xu, Y.; Zhou, G.; Liu, M. (2012). A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem, *Computers & Industrial Engineering*, Vol. 62, No. 4, 917-926, doi:[10.1016/j.cie.2011.12.014](https://doi.org/10.1016/j.cie.2011.12.014)
- [16] Lerher, T.; Ekren, Y. B.; Sari, Z.; Rosi, B. (2015). Simulation analysis of shuttle based storage and retrieval systems, *International Journal of Simulation Modelling*, Vol. 14, No. 1, 48-59, doi:[10.2507/IJSIMM14\(1\)5.281](https://doi.org/10.2507/IJSIMM14(1)5.281)
- [17] Mahnam, M.; Moslehi, G. (2009). A branch-and-bound algorithm for minimizing the sum of maximum earliness and tardiness with unequal release times, *Engineering Optimization*, Vol. 41, No. 6, 521-536, doi:[10.1080/03052150802657290](https://doi.org/10.1080/03052150802657290)
- [18] Wang, X.; Gao, L.; Zhang, C.; Shao, X. (2010). A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, Vol. 51, No. 5-8, 757-767, doi:[10.1007/s00170-010-2642-2](https://doi.org/10.1007/s00170-010-2642-2)
- [19] Cao, X.; Yang, Z. (2011). An improved genetic algorithm for dual-resource constrained flexible job shop scheduling, *4<sup>th</sup> International Conference on Intelligent Computation Technology and Automation*, Vol. 1, 42-45, doi:[10.1109/icicta.2011.18](https://doi.org/10.1109/icicta.2011.18)
- [20] Wang, S. Y.; Wang, L.; Xu, Y.; Liu, M. (2013). An effective estimation of distribution algorithm for the flexible job-shop scheduling problem with fuzzy processing time, *International Journal of Production Research*, Vol. 51, No. 12, 3778-3793, doi:[10.1080/00207543.2013.765077](https://doi.org/10.1080/00207543.2013.765077)
- [21] Xu, Y.; Wang, L.; Wang, S. (2013). An effective shuffled frog-leaping algorithm for the flexible job-shop scheduling problem, *2013 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, 128-134, doi:[10.1109/cica.2013.6611673](https://doi.org/10.1109/cica.2013.6611673)
- [22] Klancnik, S.; Ficko, M.; Balic, J.; Pahole, I. (2015). Computer vision-based approach to end mill tool monitoring, *International Journal of Simulation Modelling*, Vol. 14, No. 4, 571-583, doi:[10.2507/IJSIMM14\(4\)1.301](https://doi.org/10.2507/IJSIMM14(4)1.301)