

# AN IMPROVED GENETIC ALGORITHM FOR RESOURCE-CONSTRAINED FLEXIBLE JOB-SHOP SCHEDULING

Wei, F. F.<sup>#</sup>; Cao, C. Y. & Zhang, H. P.

School of Management and Economics, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

E-Mail: 1255699089@qq.com (<sup>#</sup> Corresponding author)

## Abstract

Flexible job-shop scheduling could effectively lower the costs of manpower and materials. However, there is little report on the scheduling algorithm or optimization model for the optimization of production resources. This paper proposes a resource-constrained flexible job-shop scheduling algorithm based on an improved genetic algorithm. Firstly, an optimization model was established for resource-constrained FJSP, together with the objective functions about resources and time, as well as constraints. Next, the multi-objective genetic algorithm (MOGA) was combined with the whale optimization algorithm (WOA) into a combinatory method to solve the proposed model for resource-constrained FJSP. Experimental results show that the combination enhances the adaptivity of crossover and mutation probabilities, and improves the local search ability, presenting an effective solution to the FJSP.

(Received in December 2020, accepted in February 2021. This paper was with the authors 1 month for 1 revision.)

**Key Words:** Multi-Objective Genetic Algorithm (MOGA), Resource Constraints, Flexible Job-Shop

## 1. INTRODUCTION

For modern manufacturing enterprises, the most important means to enhance competitiveness and gain an advantage in commercial competition include reducing the makespan and lowering the production cost [1-4]. Relying on personal experience, the traditional job-shop scheduling principles like first in, first out (FIFO), and shortest job first have not achieved the ideal effect [5, 6]. Thus, the solution to the flexible job-shop scheduling problem (FJSP) has attracted wide attention at home and abroad [7-10].

The current research on FJSP mainly deals with the algorithm design and optimization of the scheduling problem, and the analysis and modelling of manufacturing energy consumption [11-15]. Wang et al. [16] set the minimal maximum makespan as the objective of job-shop scheduling, proposed a 5/3 approximate sorting algorithm and the lower limit of the optimal solution, and discussed the properties of the optimal solution of the order-constrained graph.

Largescale customized production involves complex operations and machines [17, 18]. Morinaga et al. [19] improved the particle swarm optimization (PSO) based on Bayesian network, which satisfies the needs of solving largescale FJSPs. Kurniawati and Nugroho [20] took the minimization of maximum makespan, standby energy consumption, and power cost as the optimization goals, constructed a fuzzy opportunity-constrained programming model for remanufacturing job-shops, and solved dynamic events like machine failure.

At this stage, there are relatively few studies on the comprehensive scheduling algorithm of the FJSP [21, 22]. For the multi-machine job-shop scheduling problem, Hammami and Stützle [23] presented an operation priority sorting strategy and a machine-driven strategy with delay constraints, which effectively increases the parallelism of processing supply and demand. Viana et al. [24] put forward a waiting strategy that suspends eligible jobs based on batch optimization. Kundakcı and Kulak [25] constructed the objective function of the batch scheduling, established the corresponding fuzzy mathematical model, and solved the batch scheduling problem based on the hybrid meta-heuristic algorithm.

Many have attempted to optimize the energy consumption of single strategy optimization and integrated optimization. For instance, Zhu et al. [26] proposed a linear optimization

algorithm for operation planning, while Zhou et al. [27] developed a genetic algorithm-based method for optimizing the idle energy consumption of machines. The existing evolutionary solution algorithms for FJSP mainly optimize the running speed and optimization efficiency via genetic operations, encoding and decoding, and fitness setting. But there is little report on the scheduling algorithm or optimization model for production resources.

This paper proposes a resource-constrained flexible job-shop scheduling algorithm based on an improved genetic algorithm, which combines the multi-objective genetic algorithm (MOGA) with the whale optimization algorithm (WOA). Experimental results show that our algorithm enhances the adaptivity of crossover and mutation probabilities, and improves the local search ability.

## 2. CONSTRUCTION OF OPTIMIZATION MODEL

### 2.1 Resource and time constraints

The production resources of FJSP can be divided into renewable resources, partially renewable resources, non-renewable resources, and restricted resources. Despite having an upper limit on supply, the renewable resources are supplemented as the operations are implemented. Let  $RR_s^U$  be the upper limit on the supply of type  $s$  renewable resource, and  $DR_{is}^U$  be the demand for type  $s$  renewable resource in the  $i^{\text{th}}$  operation. Then, the constraint of the renewable resource is:

$$\sum_{i \in P_t} DR_{is}^U \leq RR_s^U, \forall s, t \quad (1)$$

where,  $P_t$  is the set of operations being implemented in period  $t$ . The total amount of non-renewable resources is determined at the start of the production in the flexible job-shop. But the non-renewable resources will not be supplemented as the operations are implemented continuously. Let  $NRR_v^T$  be the total amount of type  $v$  non-renewable resource, and  $DNR_{iv}^T$  be the demand for type  $v$  non-renewable resource in the  $i^{\text{th}}$  operation. Then, the constraint of the non-renewable resource is:

$$\sum_{v=1}^V DNR_{iv}^T \leq NRR_v^T, \forall v \quad (2)$$

For resource-constrained FJSP, the optimization is often constrained by the time window, that is, the start time of an operation is limited in a fixed time range. Let  $\Delta t_{ij}^{\min}$  and  $\Delta t_{ij}^{\max}$  be the minimum and maximum differences between the start times of the  $i^{\text{th}}$  and the  $j^{\text{th}}$  operations, respectively. Then, the time window constraint is:

$$s_i + \Delta t_{ij}^{\min} \leq s_j \leq s_i + \Delta t_{ij}^{\max} \quad (3)$$

where,  $s_i$  and  $s_j$  are the start times of the  $i^{\text{th}}$  and the  $j^{\text{th}}$  operations, respectively.

### 2.2 Resource and time objective functions

Let  $COST_s(RR_s)$  be the cost function of type  $s$  production resource, with  $RR_s$  be the replenishment of type  $s$  renewable resource. Then, the cost minimization function is:

$$\min \sum_{s=1}^S COST_s(RR_s) \quad (4)$$

The total demand for type  $s$  resource in period  $t$  is:

$$RD_{st} = \sum_{i \in P_t} DR_{is} \quad (5)$$

The mean demand for the type of resource in the production cycle is:

$$\overline{RD}_s = \frac{\sum_{t=1}^{T_T} RD_{st}}{T_T} \tag{6}$$

where,  $T_T$  is the total time of the production cycle. The minimal resource cost can be defined as:

$$\min \sum_{t=1}^{T_T} (RD_{st} - \overline{RD}_s) \tag{7}$$

If multiple kinds of production resources need to be scheduled throughout the production cycle, different weights should be assigned to each type of resource. Thus, the objective function can be written as:

$$\min \sum_{s=1}^S \omega_s \sum_{t=1}^{T_T} (RD_{st} - \overline{RD}_s)^2 \tag{8}$$

As for the objective function of time, this paper chooses four objectives for optimizing the resource-constrained FJSP, namely, makespan, makespan expectation, operation delay, and cycle delay. Let  $e_i$  be the completion time of the  $i^{\text{th}}$  operation. Then, the minimization of the makespan is:

$$\min \sum_{i=1}^{T_T} e_i \tag{9}$$

The minimization of makespan expectation, that is, minimizing the difference between the actual completion time  $e_i$  and the latest completion time  $e'_i$  of an operation, can be defined as:

$$\min |e_i - e'_i| \tag{10}$$

After assigning weights to different operations by importance, the minimization of the weighted operation delay is:

$$\min \sum_{i=1}^M \omega_i \max\{0, e_i - e'_i\} \tag{11}$$

The minimization of the cycle delay is:

$$\min \max \left\{ 0, \sum_{i=1}^{T_T} e_i - e' \right\} \tag{12}$$

where,  $e'$  is the latest completion time of the entire production cycle.

### 2.3 Mathematical modelling of scheduling

As shown in Fig. 1, the optimization of resource-constrained FJSP is a directed graph  $DG = (P, R)$ . Let  $M$  be the number of operations, and  $P$  be the set of operations. The start time, completion time, and makespan of the  $i^{\text{th}}$  operation satisfies the equation  $e_i = s_i + \Delta e_i$ . Fig. 2 shows the priority relations of four kinds of operations: start-start, end-start, end-end, and start-end. Then, the set of priority relations  $R$  in the directed graph  $DG$  can be established.

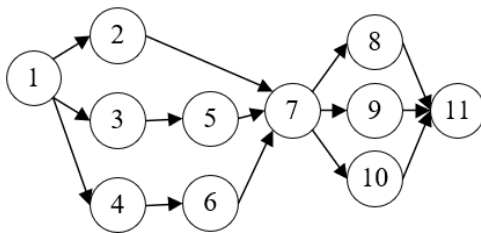


Figure 1: Structure of a node-type directed graph network.

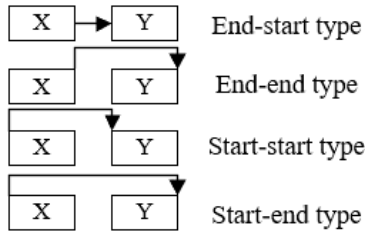


Figure 2: Priority relations of four kinds of operations.

For the optimization model for resource-constrained FJSP, the minimization of production cycle can be defined as:

$$\text{OF1: } \min \sum_{t=1}^{T_r} e_i \quad (13)$$

Eq. (13) needs to satisfy the following constraints:

$$\begin{cases} s_i \geq \max_{r \in R_i} e_r, \forall i \\ \sum_{i \in P_j} DR_{is}^U \leq RR_s^U, \forall s, t \end{cases} \quad (14)$$

Eq. (14) specifies that all the operations that have priority relations with the  $j^{\text{th}}$  operation must be completed earlier than the start time of the  $j^{\text{th}}$  operation, and the demand of this operation for a renewable resource must be lower than the upper limit on the supply of that resource. Hence, a decision variable can be constructed as:

$$\delta_{it} = \begin{cases} 1, & \text{if the } i^{\text{th}} \text{ operation is completed in period } t \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

The optimization model for resource-constrained FJSP can be updated as:

$$\text{OF2: } \min \sum_{t=1}^{T_r} e_i \quad (16)$$

Eq. (16) needs to satisfy the following constraints:

$$\begin{cases} \sum_{t=1}^{T_r} (t - \Delta e_i) \delta_{it} - \sum_{t=1}^{T_r} t \cdot \delta_{rt} \geq 0, \forall r \in R_i, \forall j \\ RR_v - \sum_{v=1}^V NDR_{iv}^T \sum_{\lambda=t}^{t+\Delta e_i-1} \delta_{i\lambda}, \forall s, t \\ \delta_{it} \in \{0, 1\}, \forall i, t \end{cases} \quad (17)$$

Eq. (17) limits the start time of each operation, the demand for renewable resources, and the feasible range of the decision variable.

### **3. PROBLEM SOLVING**

The current MOGA can effectively shorten the time of problem-solving, while outputting diverse solutions. However, the algorithm is, to a certain extent, limited in the adaptation to crossover and mutation probabilities, and the local search ability. This paper combines MOGA with WOA to solve the resource-constrained FJSP.

Fig. 3 explains the coding method of the MOGA improved for solving the resource-constrained FJSP. Let  $CP$  and  $MP$  be the crossover and mutation probabilities of MOGA, respectively. Then, the  $CP$  and  $MP$  values were adjusted based on fitness. Through dynamic control of crossover and mutation operators, the optimal individuals with high fitness were preserved to obtain the set of pareto optimal solutions.

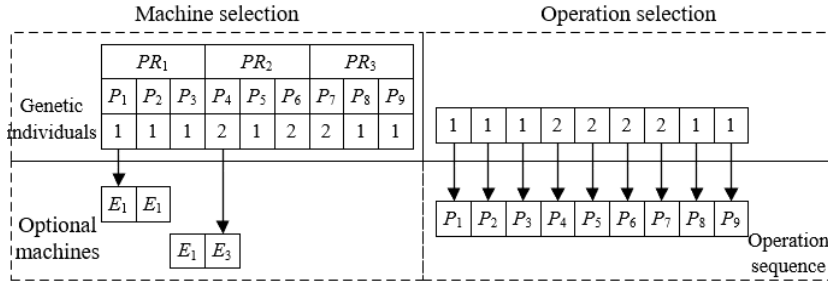


Figure 3: Coding method of improved MOGA.

This paper improves the fixed  $CP$  and  $MP$  into adaptive  $CP$  and  $MP$ . Specifically, the maximums ( $CP_{max}$  and  $MP_{max}$ ) and minimums ( $CP_{min}$  and  $MP_{min}$ ) of  $CP$  and  $MP$  were selected, and the adaptive  $CP$  and  $MP$  would change adaptively to fitness. Let  $Fitness_i$  be the fitness of the  $i^{th}$  individual, and  $Fitness_{max}$  and  $Fitness_{min}$  be the maximum and minimum fitness, respectively. Then, the crossover probability of the  $i^{th}$  individual can be computed by:

$$CP_i = CP_{min} - (Fitness_i - Fitness_{min}) / (\varepsilon + (Fitness_{max} - Fitness_{min})) * (CP_{max} - CP_{min}) \quad (18)$$

where,  $\varepsilon$  is a small real number that prevents the denominator from being zero. The mutation probability of the  $i^{th}$  individual can be calculated by:

$$MP_i = MP_{min} - (Fitness_i - Fitness_{min}) / (\varepsilon + (Fitness_{max} - Fitness_{min})) * (MP_{max} - MP_{min}) \quad (19)$$

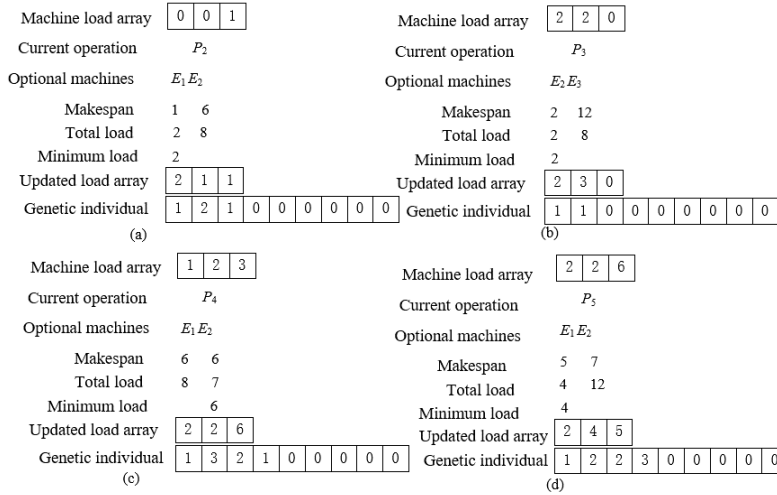


Figure 4: Generation of the new population based on elite retention strategy.

As shown in Eqs. (18) and (19), if an individual has a higher fitness than other individuals in the population, then it will have small  $CP_i$  and  $MP_i$ , that is, it is difficult to undergo crossover and mutation. To avoid damages to the set of optimal solutions, such an individual was directly preserved to the next generation. Fig. 4 explains the generation of the new population.

The WOA consists of three phases, namely, encircling prey, bubble-net attacking, and search for prey. Let  $W = (W_1, W_2, \dots, W_N)$  be the position of each whale in an  $N$ -dimensional solution space. For each whale, the probability of encircling prey is equal to that of bubble-net attacking:  $EP = AP = 0.5$ . Based on the principle of best search agency, it was assumed that the position from which the  $i^{th}$  whale chooses to swim towards the target prey as the position of the whale closest to the prey  $P_{best}$ . Then, the position of the  $i^{th}$  whale can be updated by:

$$P_i^{t+1} = P_{best}^t - A \cdot |C \cdot P_{best}^t - P_i^t| \quad (20)$$

where,  $t$  is the number of current iteration;  $A$  and  $C$  are coefficient vectors, in which each dimension consists of random numbers from a certain range. Suppose  $\gamma$  decreases linearly from 2 to 0 with the growth in  $t$ . Then,  $A$  and  $C$  can be calculated by:

$$\begin{cases} A = 2\gamma \cdot \mu_1 - \gamma \\ C = 2 \cdot \mu_2 \end{cases} \quad (21)$$

where,  $\mu_1$  and  $\mu_2$  are both random vectors in  $[0, 1]$ . The value of  $A$  determines when a whale chooses to swim towards the optimal target, and when it selects a target. If the modulus of  $A$  is smaller than 1, then the whale will swim towards the optimal individual; if the modulus of  $A$  is equal to or greater than 1, then the whale will swim towards a random individual. Fig. 5 shows all possible positions of the whale moving from the current position  $(CL_x, CL_y)$  to the target position  $(TL_x, TL_y)$  in the two-dimensional space, when the modulus of  $A$  is smaller than 1.

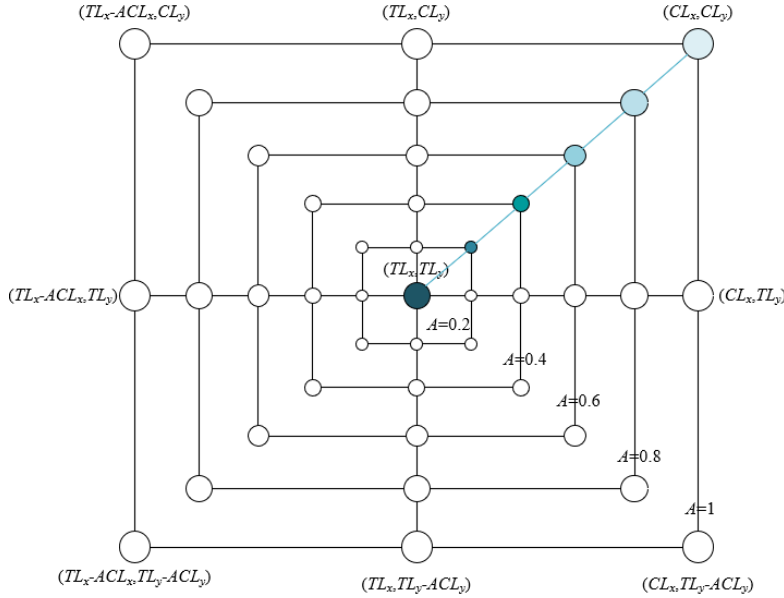


Figure 5: Positions of a whale encircling the prey.

During bubble-net attacking, the trajectory of a whale is a shrinking circle about the prey (Fig. 6). In this phase, the position of the whale can be updated by:

$$P_i^{t+1} = |P_{best}^t - P_i^t| \cdot e^{vl} \cdot \cos(2\pi l) + P_i^{t+1} \quad (22)$$

where,  $v$  is a constant defining the shape of a logarithmic spiral;  $l$  is a random number uniformly distributed in  $[-1, 1]$ . Fig. 7 shows how our algorithm solves the resource-constrained FJSP.

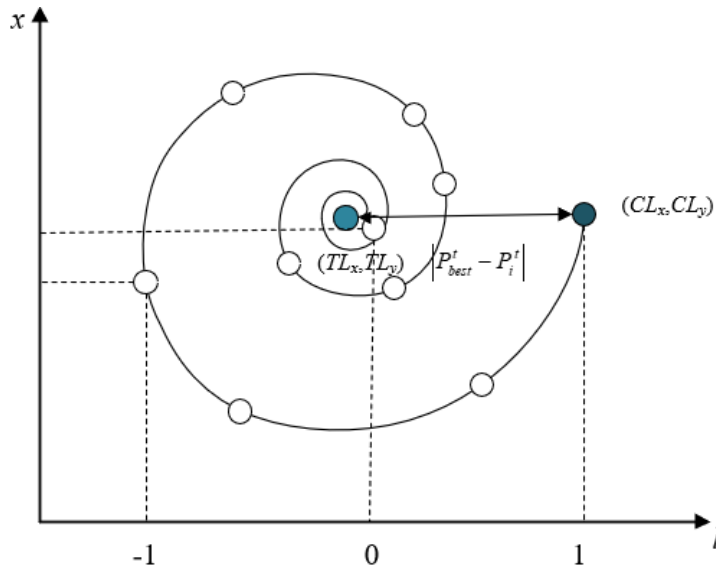


Figure 6: Trajectory of a whale in bubble-net attacking.

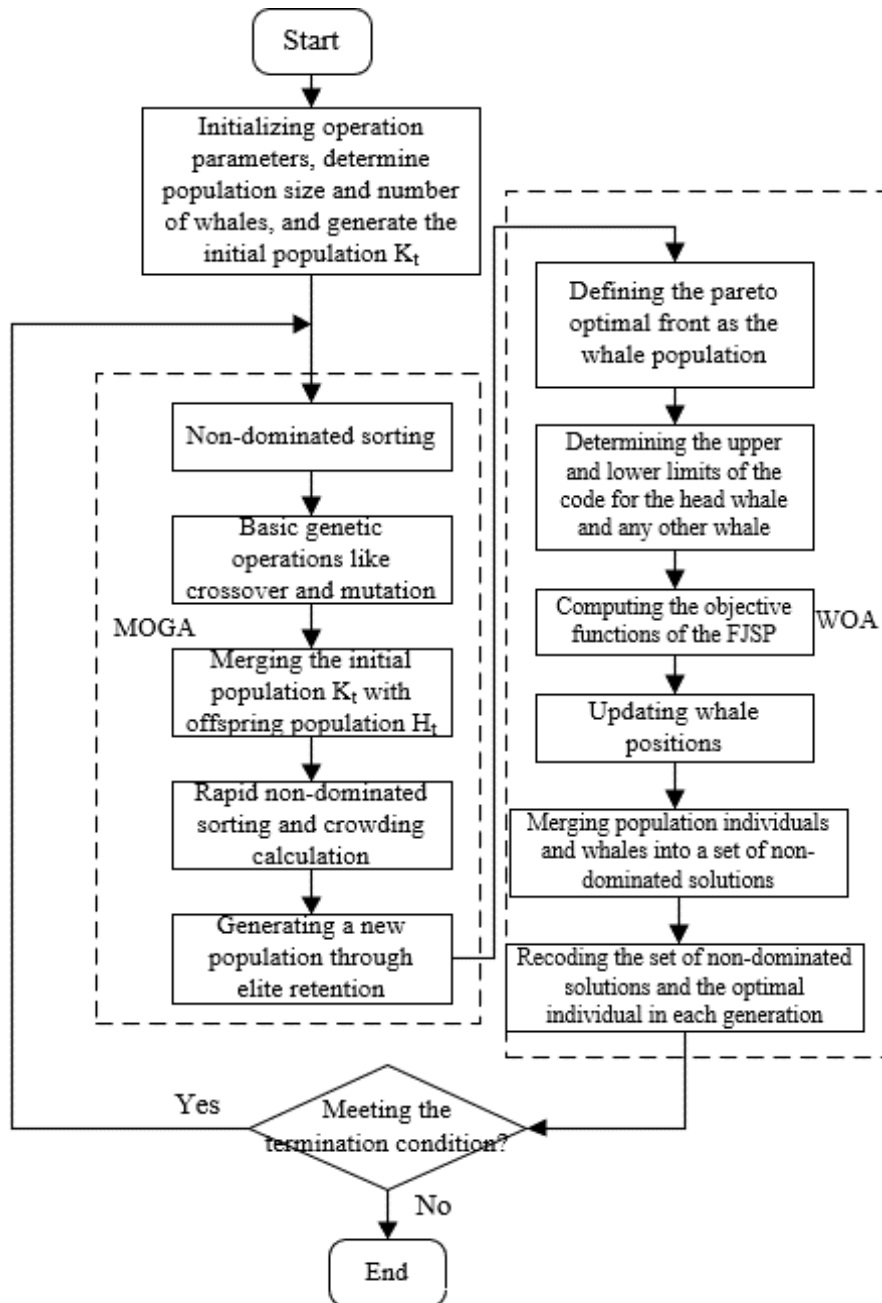


Figure 7: Flow chart of our algorithm.

#### 4. EXPERIMENTS AND RESULTS ANALYSIS

To demonstrate its effectiveness, our algorithm for resource-constrained FJSP was tested with the actual production data from the processing shop of a carmaker as the benchmark example. The example contains 15 jobs and 6 machines. The improved MOGA was compared with the original MOGA to reflect the reasonability of the proposed hybrid algorithm. As shown in Fig. 8 improved algorithm had a certain advantage in both optimization speed and the quality of the initial population.

Next, our algorithm was subject to a comparison experiment under different pareto optimal fronts. The initial population size, number of genetic iterations, initial number of whales, and number of generations were all set to 100; the maximum and minimum crossover probabilities were set to 0.6 and 0.9, respectively; the maximum and minimum mutation probabilities were set to 0.06 and 0.25, respectively.

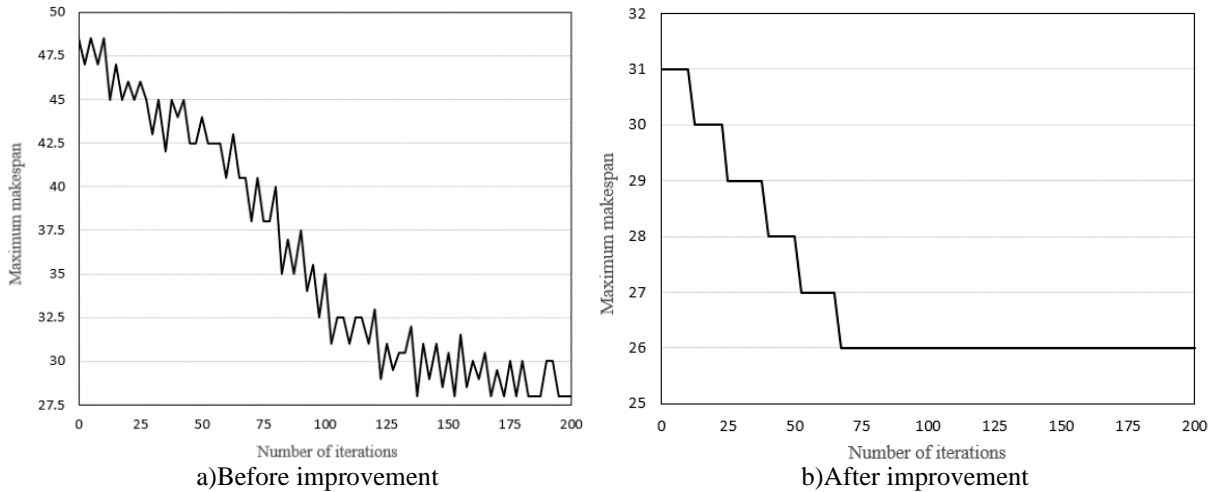


Figure 8: Maximum makespan curves obtained by the original and improved MOGAs.

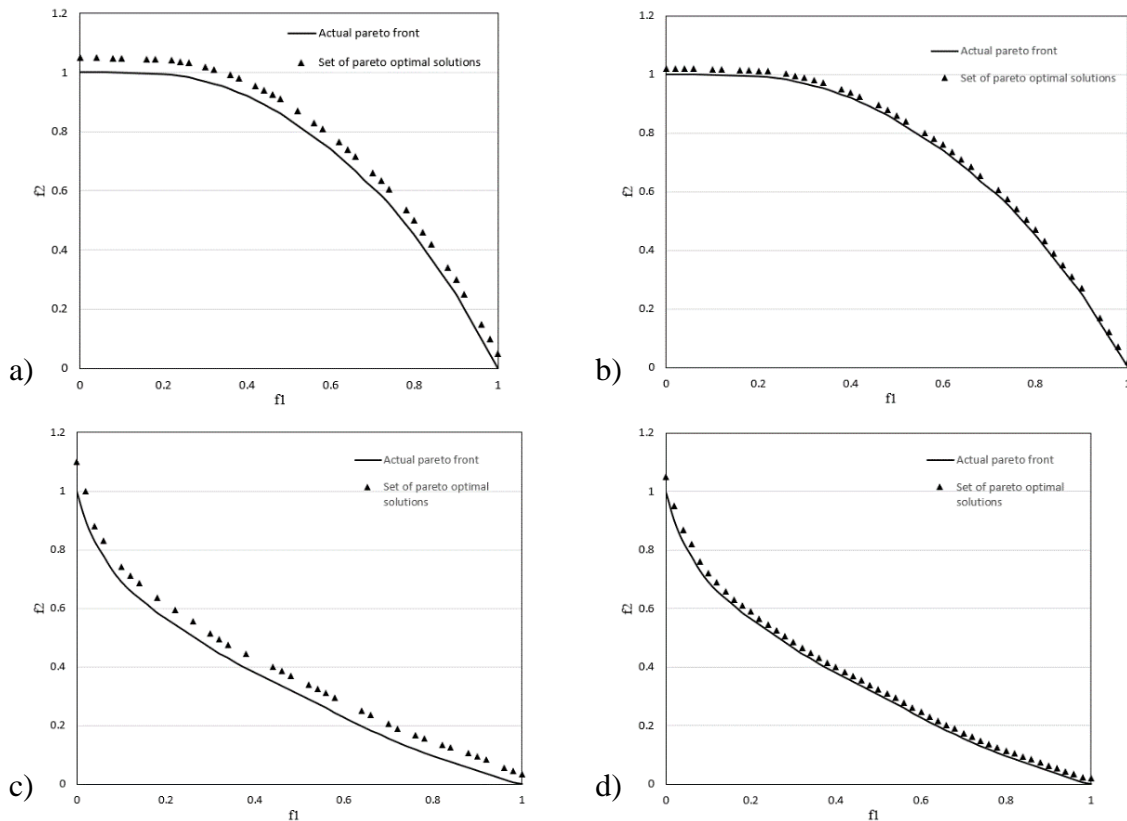


Figure 9: Performance comparison between original and improved MOGAs.

As shown in Figs. 9 a and 9 b, after 100 iterations, the set of pareto optimal solutions obtained by the original algorithm was far away from the actual pareto front, while that obtained by the improved algorithm approximated that front. In Table I, the reversal distance of the improved algorithm was smaller than that of the original algorithm by 0.00087893, while the hypervolume index of the former was larger than that of the original algorithm by 0.02193. This means the improved algorithm outputs better solution set, converges to more diverse solutions, and achieves better convergence.

Without changing other parameters, the original and improved MOGAs were tested on different test functions. The relevant results are displayed in Figs. 9 c and 9 d and Table I. It can be seen that the set of pareto optimal solutions obtained by the improved algorithm was closer to the actual pareto front than that obtained by the original algorithm. In Table I, the reversal



distance of the improved algorithm was smaller than that of the original algorithm by 0.0013196, while the hypervolume index of the former was larger than that of the original algorithm by 0.04013. The results confirm the effectiveness of the improved algorithm.

Table I: Performance of original and improved algorithms.

Test number	Algorithm	Reversal distance	Hypervolume index
1	Before improvement	0.00156227	0.65276
	After improvement	0.00068334	0.67469
2	Before improvement	0.00184162	0.28140
	After improvement	0.00052196	0.32153

Table II: Optional machines and makespans.

Job	Operation	Optional machines and makespans						
		$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	$E_6$	$E_7$
1	1	3	-	4	-	2	1	2
	2	-	4	-	2	5	3	1
	3	3	-	-	2	4	2	-
2	1	-	2	-	2	3	2	-
	2	-	4	5	-	2	1	4
	3	3	-	4	1	-	5	7
3	1	4	-	3	-	2	3	3
	2	3	-	5	2	-	4	-
	3	3	-	6	4	-	8	6
4	1	3	4	5	-	4	-	7
	2	-	3	6	3	5	-	-
	3	2	-	2	2	-	6	1
5	1	-	3	-	2	5	6	-
	3	3	-	4	5	-	5	3

Based on the said benchmark example (Table II), the production data on 5 jobs, 7 machines, and 14 operations were sorted out. Without changing the parameter setting or test parameters, the proposed improved algorithm was implemented 10 times consecutively. The optimal Gantt chart is shown in Fig. 10. It can be seen that the improved algorithm output the minimum makespan at 11.75 h, shorter than that obtained by the original algorithm (13.42 h), i.e., it is superior in solving resource-constrained FJSP.

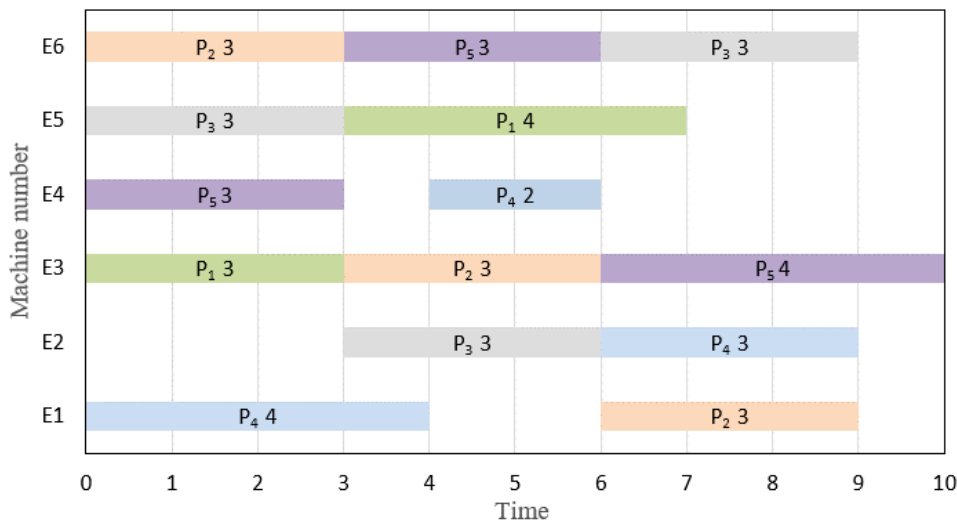


Figure 10: Gantt chart for resource-constrained FJSP.

## **5. CONCLUSIONS**

This paper designs a resource-constrained flexible job-shop scheduling algorithm based on an improved genetic algorithm. The ability of our algorithm to solve multiple objectives was tested under different pareto optimal fronts. The algorithm was found to output better solution set, converge to more diverse solutions, and achieve better convergence than the original algorithm. Besides, the optimal Gantt chart was plotted with the actual production data, which verifies the superiority of our algorithm in solving resource-constrained FJSP.

## **ACKNOWLEDGEMENTS**

This paper is supported by Postgraduate Education Reform Project of Henan Province, China (2019SJGLX045Y), and Postgraduate Education Reform and Quality Improvement Project of Henan Province, China (HNYJS2018KC02).

## **REFERENCES**

- [1] He, P. (2018). Optimization and simulation of remanufacturing production scheduling under uncertainties, *International Journal of Simulation Modelling*, Vol. 17, No. 4, 734-743, doi:[10.2507/IJSIMM17\(4\)CO20](https://doi.org/10.2507/IJSIMM17(4)CO20)
- [2] Nazifa, T. H.; Ramachandran, K. K. (2019). Information sharing in supply chain management: a case study between the cooperative partners in manufacturing industry, *Journal of System and Management Sciences*, Vol. 9, No. 1, 19-47, doi:[10.33168/JSMS.2019.0102](https://doi.org/10.33168/JSMS.2019.0102)
- [3] Liao, J.; Lin, C. (2019). Optimization and simulation of job-shop supply chain scheduling in manufacturing enterprises based on particle swarm optimization, *International Journal of Simulation Modelling*, Vol. 18, No. 1, 187-196, doi:[10.2507/IJSIMM18\(1\)CO5](https://doi.org/10.2507/IJSIMM18(1)CO5)
- [4] Ikechukwu, A. J. (2019). Assessment of organizational performance of private manufacturing companies: the impact of supply chain management responsiveness, *Journal of System and Management Sciences*, Vol. 9, No. 3, 26-44, doi:[10.33168/JSMS.2019.0302](https://doi.org/10.33168/JSMS.2019.0302)
- [5] Bustos-Tellez, C. A.; Tenjo-García, J. S.; Figueroa-García, J. C. (2018). Solving job-shop scheduling problems with fuzzy processing times and fuzzy due dates, Medina, J. (Ed.), *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, Cham, 790-799, doi:[10.1007/978-3-319-91473-2\\_66](https://doi.org/10.1007/978-3-319-91473-2_66)
- [6] Li, H.; Cao, B.; Zhu, H. (2017). A variable neighborhood migrating birds optimization algorithm for flexible job shop scheduling, *International Journal of Performability Engineering*, Vol. 13, No. 7, 1020-1029, doi:[10.23940/ijpe.17.07.p3.10201029](https://doi.org/10.23940/ijpe.17.07.p3.10201029)
- [7] Mhasawade, S. N.; Bewoor, L. A. (2017). A survey of hybrid metaheuristics to minimize makespan of job shop scheduling problem, *Proceedings of the 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, 1957-1960, doi:[10.1109/ICECDS.2017.8389792](https://doi.org/10.1109/ICECDS.2017.8389792)
- [8] Mishra, S. K.; Bose, P. S. C.; Rao, C. S. P. (2017). An invasive weed optimization approach for job shop scheduling problems, *The International Journal of Advanced Manufacturing Technology*, Vol. 91, No. 9, 4233-4241, doi:[10.1007/s00170-017-0091-x](https://doi.org/10.1007/s00170-017-0091-x)
- [9] Nouri, H. E.; Driss, O. B.; Ghédira, K. (2016). Simultaneous scheduling of machines and transport robots in flexible job shop environment using hybrid metaheuristics based on clustered holonic multiagent model, *Computers & Industrial Engineering*, Vol. 102, 488-501, doi:[10.1016/j.cie.2016.02.024](https://doi.org/10.1016/j.cie.2016.02.024)
- [10] Teekeng, W.; Thammano, A.; Unkaw, P.; Kiatwuthiamorn, J. (2016). A new algorithm for flexible job-shop scheduling problem based on particle swarm optimization, *Artificial Life and Robotics*, Vol. 21, No. 1, 18-23, doi:[10.1007/s10015-015-0259-0](https://doi.org/10.1007/s10015-015-0259-0)
- [11] Selmair, M.; Claus, T.; Trost, M.; Bley, A.; Herrmann, F. (2016). Job shop scheduling with flexible energy prices, *Proceedings of the 2016 European Council for Modeling and Simulation*, 488-494, doi:[10.7148/2016-0488](https://doi.org/10.7148/2016-0488)

- [12] Rochow, P.; Böning, C.; Prinzhorn, H. (2015). Qualitätsorientierte belegungsplanung, *Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb*, Vol. 110, No. 10, 599-602, doi:[10.3139/104.111406](https://doi.org/10.3139/104.111406)
- [13] Shen, Z.; Burnham, K. J.; Smalov, L. (2015). Optimised job-shop scheduling via genetic algorithm for a manufacturing production system, Selvaraj, H.; Zydek, D.; Chmaj, G. (Eds.), *Progress in Systems Engineering, Advances in Intelligent Systems and Computing*, Vol. 366, Springer, Cham, 89-92, doi:[10.1007/978-3-319-08422-0\\_13](https://doi.org/10.1007/978-3-319-08422-0_13)
- [14] Phanden, R. K.; Jain, A. (2015). Assessing the impact of changing available multiple process plans of a job type on mean tardiness in job shop scheduling, *The International Journal of Advanced Manufacturing Technology*, Vol. 80, No. 9, 1521-1545, doi:[10.1007/s00170-015-7123-1](https://doi.org/10.1007/s00170-015-7123-1)
- [15] Driss, I.; Mouss, K. N.; Laggoun, A. (2015). A new genetic algorithm for flexible job-shop scheduling problems, *Journal of Mechanical Science and Technology*, Vol. 29, No. 3, 1273-1281, doi:[10.1007/s12206-015-0242-7](https://doi.org/10.1007/s12206-015-0242-7)
- [16] Wang, C. L.; Rong, G.; Weng, W.; Feng, Y. P. (2015). Mining scheduling knowledge for job shop scheduling problem, *IFAC-PapersOnLine*, Vol. 48, No. 3, 800-805, doi:[10.1016/j.ifacol.2015.06.181](https://doi.org/10.1016/j.ifacol.2015.06.181)
- [17] Mokhtari, H.; Dadgar, M. (2015). Scheduling optimization of a stochastic flexible job-shop system with time-varying machine failure rate, *Computers & Operations Research*, Vol. 61, 31-45, doi:[10.1016/j.cor.2015.02.014](https://doi.org/10.1016/j.cor.2015.02.014)
- [18] Yazdani, M.; Zandieh, M.; Tavakkoli-Moghaddam, R.; Jolai, F. (2015). Two meta-heuristic algorithms for the dual-resource constrained flexible job-shop scheduling problem, *Scientia Iranica*, Vol. 22, No. 3, 1242-1257
- [19] Morinaga, E.; Nakamura, T.; Wakamatsu, H.; Arai, E. (2017). A method of flexible job-shop scheduling for highly-distributed manufacturing systems, *Proceedings of the 24<sup>th</sup> International Conference on Production Research*, 592-597, doi:[10.12783/dtetr/icpr2017/17676](https://doi.org/10.12783/dtetr/icpr2017/17676)
- [20] Kurniawati, D. A.; Nugroho, Y. I. (2017). Computational study of N-job M-machine flow shop scheduling problems: SPT, EDD, NEH, NEH-EDD, and modified-NEH algorithms, *Journal of Advanced Manufacturing Systems*, Vol. 16, No. 4, 375-384, doi:[10.1142/S0219686717500226](https://doi.org/10.1142/S0219686717500226)
- [21] Rohaninejad, M.; Sahraeian, R.; Nouri, B. V. (2017). Minimising the total cost of tardiness and overtime in a resumable capacitated job shop scheduling problem by using an efficient hybrid algorithm, *International Journal of Industrial and Systems Engineering*, Vol. 26, No. 3, 318-343, doi:[10.1504/IJISE.2017.084421](https://doi.org/10.1504/IJISE.2017.084421)
- [22] Yazdani, M.; Aleti, A.; Khalili, S. M.; Jolai, F. (2017). Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 107, 12-24, doi:[10.1016/j.cie.2017.02.019](https://doi.org/10.1016/j.cie.2017.02.019)
- [23] Hammami, H.; Stützle, T. (2017). A computational study of neighborhood operators for job-shop scheduling problems with regular objectives, Hu, B.; López-Ibáñez, M. (Eds.), *Evolutionary Computation in Combinatorial Optimization*, Springer, Cham, 1-17, doi:[10.1007/978-3-319-55453-2\\_1](https://doi.org/10.1007/978-3-319-55453-2_1)
- [24] Viana, M. S.; Junior, O. M.; Contreras, R. C. (2020). An improved local search genetic algorithm with multi-crossover for job shop scheduling problem, Rutkowski, L.; Scherer, R.; Korytkowski, M.; Pedrycz, W.; Tadeusiewicz, R.; Zurada, J. M. (Eds.), *Artificial Intelligence and Soft Computing*, Springer, Cham, 464-479, doi:[10.1007/978-3-030-61401-0\\_43](https://doi.org/10.1007/978-3-030-61401-0_43)
- [25] Kundakci, N.; Kulak, O. (2016). Hybrid genetic algorithms for minimizing makespan in dynamic job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 96, 31-51, doi:[10.1016/j.cie.2016.03.011](https://doi.org/10.1016/j.cie.2016.03.011)
- [26] Zhu, H.; Deng, Q.; Zhang, L.; Hu, X.; Lin, W. (2020). Low carbon flexible job shop scheduling problem considering worker learning using a memetic algorithm, *Optimization and Engineering*, Vol. 21, No. 4, 1691-1716, doi:[10.1007/s11081-020-09494-y](https://doi.org/10.1007/s11081-020-09494-y)
- [27] Zhou, G.; Zhou, Y.; Zhao, R. (2021). Hybrid social spider optimization algorithm with differential mutation operator for the job-shop scheduling problem, *Journal of Industrial & Management Optimization*, Vol. 17, No. 2, 533-548, doi:[10.3934/jimo.2019122](https://doi.org/10.3934/jimo.2019122)