Whale optimization algorithm with opposition-based learning strategy for solving flexible job shop scheduling problem

Mingliang Wu[1,2,*], *Dongsheng* Yang[1,2], and *Tianyi* Liu[1,2]

[1]Intelligent Electrical Science and Technology Research Institute, Northeastern University, Liaoning Shenyang, 110819, China
[2] Key Laboratory of Integrated Energy Optimization and Secure Operation of Liaoning Province, Northeastern University, Liaoning, Shenyang, 110819, China

Abstract. Flexible job shop scheduling problem is the allocation of available shared resources and the sequencing of processing tasks within a certain period of time to meet certain or certain specific production indicators. The research and application of effective scheduling methods and optimization technologies are the foundation and key to realizing advanced manufacturing and improving production efficiency. Improving the production scheduling plan can greatly improve production efficiency and resource utilization, thereby enhancing the competitiveness of enterprises. Therefore, the production scheduling problem has always been a research hotspot in manufacturing systems. In this paper, we introduce the opposition-based learning strategy and combine it with whale optimization algorithm to solving flexible job shop scheduling problem better. 10 FJSP cases are introduced to test the performance of our algorithm and other comparison algorithms. The results obtrain show that our algorithm is more better and practical than other algorithm when dealing with FJSP cases.

Keywords: Whale optimization algorithm, Flexible job shop scheduling problem, Makespan, Opposition-based learning strategy.

1 Introduction

Manufacturing is a pillar industry of a country and an important driving force for the growth of the national economy [1]. In recent years, with the rapid development of information technology and the increasingly fierce global competition, market demand has shown a diversified trend, which has brought new requirements and challenges to the manufacturing field [2]. In order to ensure the efficient operation of enterprises and reduce production costs, it is necessary to use more reasonable and effective scheduling schemes in the production process [3]. In addition, the flexible job shop scheduling problem (FJSP) has been proved to be a kind of NP-hard problem [4]. So far, people have not found an

* Corresponding author: wuml_neu@163.com

algorithm with the complexity of polynomial time to effectively solve this kind of problem. Therefore, the FJSP has been extensively studied by many scholars from related fields since it was put forward [5]. The traditional operations research method first uses mathematical modeling of the problem, and then uses operations research related methods to solve the problem [6]. Among the more common methods are dynamic programming algorithm, Laplace relaxation method, linear programming method is equivalent. The above-mentioned algorithms are effective in solving small-scale problems and can often obtain satisfactory results. However, as the scale of the problem to be solved increases, traditional operations research methods are no longer suitable for solving such problems [7].

In recent years, with the continuous development and improvement of optimization theory, many swarm intelligence optimization algorithms based on natural heuristics have been continuously proposed and applied to related job shop scheduling fields, including genetic algorithm (GA), artificial bee colony algorithm (ABC) , ant colony algorithm (ACO), migratory bird algorithm (MBO), whale swarm optimization algorithm (WOA), particle swarm optimization optimization algorithm (PSO) and other swarm intelligence algorithms to solve the problem by imitating the way of cooperation between organisms in nature optimized solution [8-13]. Swarm intelligence optimization algorithm is relatively easy to implement, robust, adaptable, not subject to specific problems, and has strong versatility. Generally speaking, it is an important method and research approach to solve the FJSP that the optimal solution or a better feasible solution to be solved in a reasonable time can be obtained [14].

In summary, the research on FJSP not only has great practical value in actual production and self-manufacturing, but also has important theoretical significance. Although many scholars have studied and explored such problems for a long time, due to the extremely high complexity of the FJSP, the complicated constraints, the high dimension of the solution space and the limitation of the actual production environment, the mathematics of the problem Models become more complex, and traditional operations research methods have been unable to solve the problem or the time required to solve the problem is too long to lose its practical value. Therefore, it is of great significance to use swarm intelligence optimization algorithm to model and solve the problem of shop scheduling. Based on the above background, this article researches and explores the FJSP. In this paper, we utilize WOA to solve practical FJSP. In oder to enhance the sloving ability of WOA, we introduce the opposition-based learning strategy (OBL) and combine it with WOA. The rest of this paper are arranged as follow. First, introduce FJSP in Section 2. Then the basic process of WOA and the details of OBL are introduced in Section 3. Then, perform experiments and conduct experimental analysis in Section 4. Finally, the full text is summarized in Section 5.

2 The FJSP

As an important field of optimization scheduling technology research, FJSP is the key basic technology module of job shop production line [15]. The definition of job scheduling is that the manufacturing job shop has multiple machines with different functions [16]. The parts to be processed are composed of multiple operation. Each operation can only be processed on one machine at the same time. Therefore, the processing routes of the parts have a mutually restrictive relationship [17]. The main goal of solving this problem is to specify the machine used in each process and the order of use, and obtain the optimal solution of these two problems under constraint conditions. For business managers, production cost and processing time are two important performance indicators that are widely considered in the production scheduling optimization problem. Without loss of generality, FJSP can be described systematically as: n jobs $J\_set = \{J_1, J_2, \dots, J_n\}$ is processing on m machines $M\_set = \{M_1, M_2, \dots, M_m\}$ in a production job shop, each job consists of uncertain operation, reasonably plan the work sequence of the job on the machine to achieve the required processing requirements and goals. The mathematics of FJSP can be expressed by a combination of constraints and optimization goals. The constraints of FJSP can be elaborated as follow:

1) There can be multiple machines for each process, but only one machine can be selected for processing.

$$\sum_{k=1}^{m} x_{ijk} = 1$$

where x_{ijk} means that whether *jth* operation of job J_i select machine M_k for its processing, if yes, it will be 1, otherwise it will be 0, *m* represents the number of machines.

2) A machine can only have one job processed on it at the same time.

$$f_{ij} < b_{i'j'}$$

where f_{ij} denotes the completion time of the jth operation of J_i. $b_{i'j'}$ denotes the begin time of following job.

3) Operations of one same job exist sequence.

$$f_{ij} \leq b_{i(i+1)}$$

4) All jobs can be processed and all machines are optional when task starts.

$$b_{ijk} \geq 0$$
$$f_{ijk} \geq 0$$

5) The processing progress is non-preemptive, and any preemptions or cancellations are not permitted.

$$b_{ijk} + w_{ijk} = f_{ijk}$$

where w_{ijk} implies the processing time of *jth* operation of J_i on machine M_k.

Most researches on FJSP focus on the main objectives: completion time, machine load, delay time, energy consumption, Etc. This paper applies the maximum completion time as the criterion for judging the value of the algorithm, which can be described as:

$$f = minC_{max} = min(maxF_i),\ i=1,2,...,n$$

where *n* represents the number of jobs

3 Whale optimization algorithm

In the whale algorithm, the position of each whale represents a feasible solution. During the hunting process of a school of whales, each whale has two behaviors. One is to surround the prey, and all the whales move toward other whales; the other is a steam drum net, where the whales swim in a circular motion and eject bubbles to drive away the prey. In each generation of swimming, whales will randomly choose these two behaviors to hunt. In the behavior of the whale to surround its prey, the whale will randomly choose whether to swim toward the best position of the whale or randomly choose a whale as its target and approach it. Suppose the position of each whale in the D-dimensional solution space is: $X = (x_1, x_2, \dots, x_D)$.

3.1 Mathematical Model of Surrounding Prey Stage

When surrounding prey, the whale will choose to swim towards the best position of the whale or towards a random whale.

3.1.1. Whale swimming towards the optimal position

The formula for updating the position of the whale is as follows:

$$X_i^{t+1} = X_{best}^t - A|C * X_{best}^t - X_i^t|$$

where X_{best}^t represents the position of the optimal whale, A is a random number uniformly distributed in (-a, a) in each dimension, the initial value of a is 2, and it linearly decreases to 0 with the number of iterations, C is a random number uniformly distributed in (0,2).

3.1.2. Swimming towards the location of a random whale

The formula for updating the position of the whale is as follows:

$$X_i^{t+1} = X_{rand}^t - A|C * X_{rand}^t - X_i^t|$$

where X_{rand}^t represents the position of the random whale. The value of A determines whether the current whale is swimming towards the optimal whale or a random whale. If $|A < 1|$, the whale chooses to swim towards the optimal individual, otherwise, the whale chooses to swim towards random individuals.

It can be seen that in the process of surrounding prey, the search mode of the whale algorithm is to search in the vicinity of the optimal individual or search in the vicinity of the random individual.

3.2 Bubble net

When hunting, whales will eject steam drums to form a bubble net to drive away their prey.

In order to use bubble nets to drive away prey, whales will constantly update their positions. When using the bubble net, the whale's position update formula is as follows:

$$X_i^{t+1} = |X_{best}^t - X_i^t| * e^{bl} * cos(2\pi l) + X_{best}^t$$

where *b* is the constant, *l* is a random number uniformly distributed in [-1,1]. Before each action, each whale will toss a coin to decide whether to surround its prey or use a bubble net to drive it away. The flow chart of the whale algorithm is summarized in Figure 1.

3.3 Opposition-based learning strategy

OBL strategy is a way proposed recently to improve the solving ability of the algorithm. The main idea of OBL is: evaluate the current solution and its inverse solution, and choose the better solution as the individual of the subsequent generation. Assume $X_i = (x_1, x_2, \ldots, x_d)$ is a current solution in D dimension space, the corresponding opposite solution $Y_i = (y_1, y_2, \ldots, y_d)$ is calculated by the following equation:

$$y_i = a_i + b_i - x_i$$

where *i* in [1,d], a_i and b_i represent the upper and lower limits of the *ith* solution respectively. The OBL can enhance the search effectiveness and improve population diversity that has been proven to improve the search ability of the algorithms.

3.4 The improved whale optimization algorithm

This paper introduce the OBL strategy and combine it with whale optimization algorithm to solving FJSP. The procedure of improved WOA is put in Figure 1. Firstly, initialize population and other parameters. Then, perform a coin toss operation, that is, randomly generate a random number. If the random number is less than 0.5, the whale hunts by surrounding the prey, otherwise, the whale hunts by the bubble net. Next, regenerate the current individual by OBL strategy, if opposite individual is better than current induvidual, replace current individual by opposite individual.

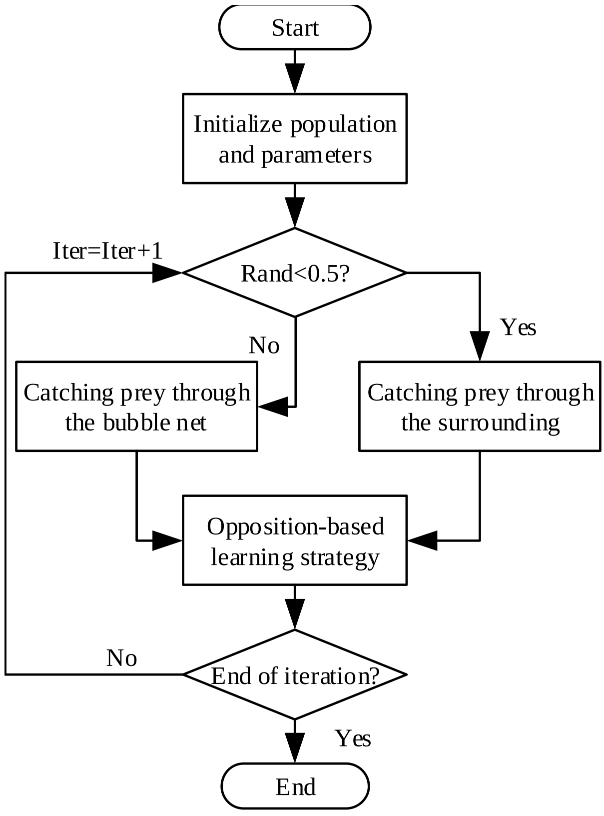


Fig. 1. The flowchart of improved WOA.

4 The experiment and discuss

This papar employ ten FJSP cases as the benchmark to examine the advantage and merit of improved WHA. Meanwhile, six other classic meta-heuristic algorithms that include: dragonfly algorithm (DA), ant colony algorithm (ACA), artificial bee colony algorithm (ABC), grey wolf algorithm (GWO), whale optimization algorithm (WOA), and grasshopper optimization algorithm (GOA) are introduced as the comparison algorithm to reflect the power of our algorithm. The maximum number of iterations and population size of all algorithms are the same, and they are also 100. The other parameters of the algorithm are set according to the suggested values in the original paper of the algorithm. In order to eliminate the randomness to the greatest extent, ten experiments were performed separately, and the average value obtained is recorded in Table 1. From Table 1, we can see that improve WOA outperform other comparison algorithm. Improved WOA can always be optimal no matter in any of the ten cases except case 2. And there is a big gap between our algorithm and other comparison algorithms. It again shows that the power and advantage of our algorithm in dealing with partical FJSP.

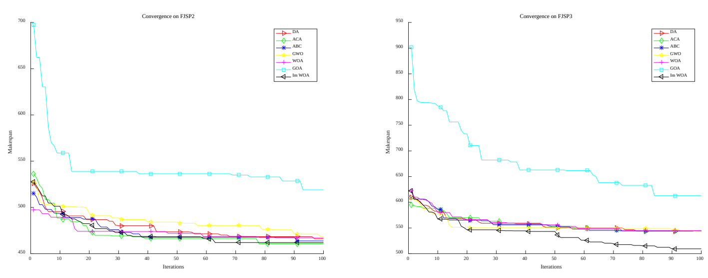
The convergence cruves of all algorithm are placed in Figure 2, we can see that our algorthim not only acheives the optimal in the end, but also keeps the absolute lead during entire iteration. It means that our algorithm both possess the strong local and global searching ability. These two abilities are both important for any meta-heuristic algorigthm, our improvment strategy can enhance original WOA effectively. As a result, we can draw a conclusion that improved WOA is more effective than other six meta-heuristic algorigthms, it is value and meaningful for this paper to introduce OBL strategy and utilize improved WOA to slove actual FJSP.

The Gantt chart of FJSP1 and FJSP10 by improved WOA are put in Figure 3 (the horizontal axis represents the completion time, and the vertical axis represents the machine number). For example, in part(a) of Figure 3, at zero time, machine one starts the first process the first operation of the third job, machine two starts the first process the first operation of the fourth job, and machine three starts the first operation of the first job. While waiting for the third job's first operation to be processed, machine four starts to process the second operation of the third job. Waiting for the first operation of the fourth job to be processed, machine five starts to process the second operation of the fourth job, and so on. The Gantt chart in Figure 3 has very few gaps, which proves that the scheduling scheme obtained by our WOA is efficient and the best choice with minimum completion time.completion time.

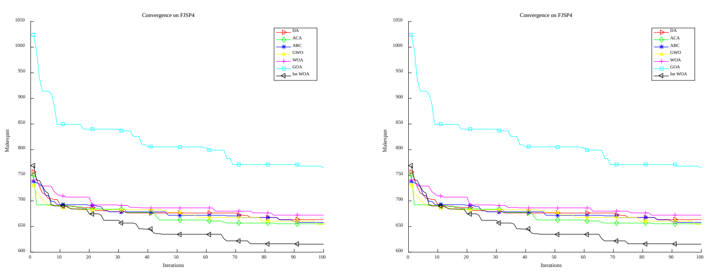
Table 1. The average results of ten counts of experiments.

Algorithm	Problem									
	FJSP1	FJSP2	FJSP3	FJSP4	FJSP5	FJSP6	FJSP7	FJSP8	FJSP9	FJSP10
DA	485.2	466	543.8	663.6	630.4	744.4	1102.6	1105.2	1349.4	1597
ACA	483.8	460.8	544.2	655.4	636.2	767.2	1113.4	1106.2	1369.4	1593.6
ABC	484.2	463.8	544.4	658.2	634.4	745	1104.2	1123.6	1385.6	1587.8
GWO	485	467	545	672.2	618.6	771.6	1084.2	1134.4	1376	1587.6
WOA	542.6	519	612.8	764.2	747	882.8	1247	1239.4	1498.8	1781.4
GOA	484.2	468.6	543.4	654.2	623	760.6	1111.2	1122.4	1335.4	1601.2
Im WOA	478.4	462	509.4	615.6	592.4	693.4	1010.2	1007.2	1262.8	1456.6

*Note: Im WOA means the improved whale optimization algorithm

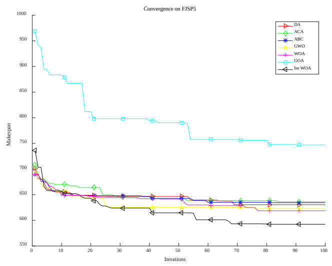


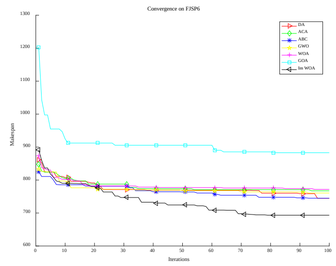
(a)FJSP1 (b)FJSP2



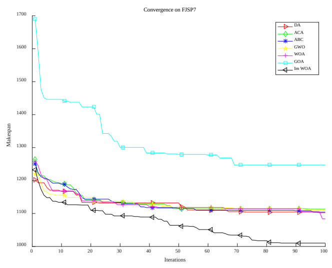
 https://doi.org/10.1051/itmconf/20224501033

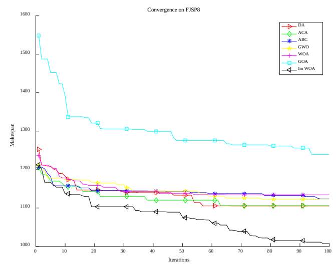
(c)FJSP3 (d)FJSP4



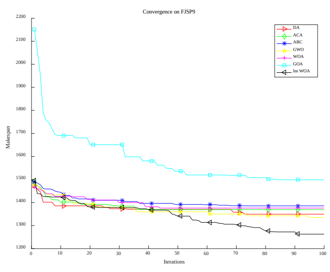


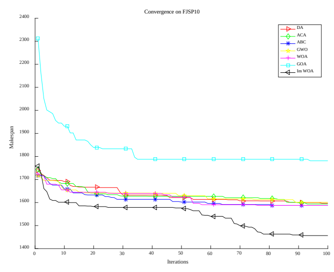
(e)FJSP5 (f)FJSP6





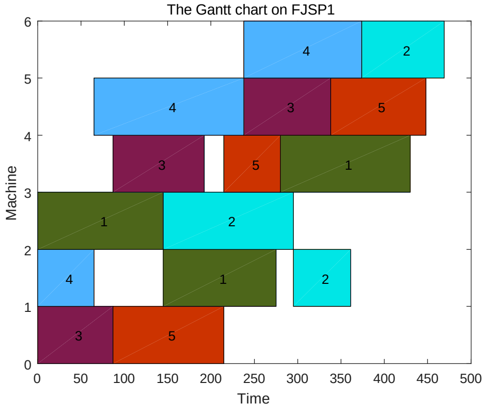
(g)FJSP7 (h)FJSP8

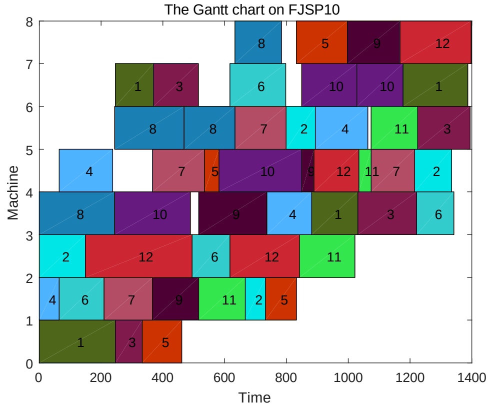




(l)FJSP9 (j)FJSP10

Fig. 2. The convergence cruve of algorithm.





(a) FJSP1 (b) FJSP10

Fig. 3. The Gantt chart on FJSP1 and FJSP10 by our algorithm.

5 Conclusion

This paper takes FJSP as the main research content, and conducts a more in-depth study and further optimization of JSP and FJSP. The modeling of FJSP is complex, the amount of calculation is large, and the complexity of the problem is high. The result of production scheduling directly affects the production efficiency and competitiveness of the entire production enterprise. To this end, whale optimization algorithm is employed to sloving the FJSP. To enhance the optimization capability of WOA, the OBL strategy is added to WOA. Extensive numerical experiments are conducted on ten FJSP cases, the average results obtained and convergence cruve of algorithm both reflect the advantage and power of our algorithm. Improve WOA can achieve the best results in comparison with other six classic meta-heuristic algorithms.

FJSP has long been a key area of research by scholars at home and abroad. Due to time constraints and insufficient practical experience, this article still has some shortcomings. On the basis of existing research, there are still the following issues worthy of further in-depth study :

1) Further study the whale optimization algorithm, establish a more rigorous mathematical theoretical basis for the algorithm, and further optimize the performance of the algorithm from the theoretical level, so that the whale optimization algorithm is more universal in solving FJSP.
2) Further study the FJSP, starting from the perspective of the problem solution space structure, in-depth analysis of the characteristics of the solution space structure of the problem, and establishing an algorithm that is more suitable for solving such problems with the solution space structure as the starting point. At the same time, it is necessary to further study the upper limit of the complexity of the problem, and determine the limit performance of the algorithm when solving the problem from a theoretical level.

This research work is supported by the National Key Research and Development Project under Grant 2018YFB1700500.

References

1. D. Lei, M. Li, and L. Wang, “A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold,” IEEE transactions on cybernetics, vol. 49, no. 3, pp. 1097–1109, 2018.
2. X. Li and L. Gao, “An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem, ” International Journal of Production Economics, vol. 174, pp. 93-110, 2016.
3. Y. Demir and S. K. isleyen, “Evaluation of mathematical models for flexible job-shop scheduling problems, ” Applied Mathematical Modelling, vol. 37, no. 3, pp. 977-988, 2013.
4. F. Pezzella, G. Morganti, and G. Ciaschetti, “A genetic algorithm for the flexible job-shop scheduling problem, ” Computers & operations research, vol. 35, no. 10, pp. 3202-3212, 2008.
5. M. Nouiri, A. Bekrar, A. Jemai, D. Trentesaux, A. C. Ammari, and S. Niar, “Two stage particle swarm optimization to solve the flexible job shop predictive scheduling problem considering possible machine breakdowns,” Computers & Industrial Engineering, vol. 112, no. oct., pp. 595–606, 2017.

6. Ham, “Flexible job shop scheduling problem for parallel batch processing machine with compatible job families,” Applied Mathematical Modelling, vol. 45, no. May, pp. 551–562, 2017.
7. H. Wang, Z. Jiang, Y. Wang, H. Zhang, and Y. Wang, “A two-stage optimization method for energy-saving flexible job-shop scheduling based on energy dynamic characterization,” Journal of Cleaner Production, vol. 188, no. JUL.1, pp. 575–588, 2018.
8. J. Li, X. Yuan, L. Bo, and Z. Zeng, “A pseudo-dynamic search ant colony optimization algorithm with improved negative feedback mechanism to solve tsp,” 2018.
9. X. Gu, M. Huang, and X. Liang, “An improved genetic algorithm with adaptive variable neighborhood search for fjsp,” Algorithms, vol. 12, no. 11, p. 243, 2019.
10. Bagheri, M. Zandieh, I. Mahdavi, and M. Yazdani, “An artificial immune algorithm for the flexible job-shop scheduling problem, ” Future Generation Computer Systems, vol. 26, no. 4, pp. 533-541, 2010.
11. S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper optimisation algorithm: theory and application, ” Advances in Engineering Software, vol. 105, pp. 30-47, 2017.
12. J. Tian-Hua, “Flexible job shop scheduling problem with hybrid grey wolf optimization algorithm, “Control and Decision, vol. 33, no. 3, pp. 503{508, 2018.
13. X. Wu, X. Liu, and N. Zhao, “An improved differential evolution algorithm for solving a distributed assembly flexible job shop scheduling problem,” Memetic Computing, vol. 11, no. 4, pp. 335-355, 2019.
14. B. Liu, L. Wang, and Y.-H. Jin, “An effective pso-based memetic algorithm for flow shop scheduling,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 1, pp. 18–27, 2007.
15. Y. Yuan, H. Xu, and J. Yang, “A hybrid harmony search algorithm for the flexible job shop scheduling problem,” Applied soft computing, vol. 13, no. 7, pp. 3259–3272, 2013.
16. Bagheri, M. Zandieh, I. Mahdavi, and M. Yazdani, “An artificial immune algorithm for the flexible job-shop scheduling problem,” Future Generation Computer Systems, vol. 26, no. 4, pp. 533-541, 2010.
17. M. Dai, D. Tang, A. Giret, and M. A. Salido, “Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints,” Robotics and Computer-Integrated Manufacturing, vol. 59, pp. 143-157, 2019.