

E-COMMERCE WORKSHOP SCHEDULING BASED ON DEEP LEARNING AND GENETIC ALGORITHM

Wu, P. J.* & Yang, D.**.#

* School of Business Administration, Anhui University of Finance and Economics,
Bengbu 233030, China

** School of Management Science and Engineering, Dongbei University of Finance and Economics,
Dalian 116025, China

E-Mail: yangdong@aufe.edu.cn (# Corresponding author)

Abstract

With the gradual rise of customized manufacturing, the connection between e-commerce and intelligence manufacturing system have been deepened, highlighting the importance of intelligent scheduling to both intelligent manufacturing and e-commerce. The key to intelligence manufacturing lies in workshop scheduling. This paper optimizes the genetic algorithm (GA) with deep learning neural network (DLNN) and applies the optimized GA to realize intelligent workshop scheduling. Firstly, the production methods of e-commerce products were analysed, as well as the features of workshop scheduling problem (WSP). On this basis, the authors established a mathematical model of the WSP. Considering the actual needs of the workshop, an integrated scheduling algorithm was designed combining DLNN and GA. The algorithm improves the GA with a DLNN called long short-term memory network (LSTM) and constructs the fitness function in a novel manner. Simulation results show that our algorithm can avoid the local optimal trap that plagues the original GA, and better the global search performance.

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

Key Words: Workshop Scheduling, Genetic Algorithm (GA), Deep Learning Neural Network (DLNN), E-Commerce, Long Short-Term Memory Network (LSTM)

1. INTRODUCTION

Artificial intelligence (AI), a crucial technique in Internet Plus manufacturing, has been quickly and efficiently applied to production decision-making of intelligence manufacturing. In manufacturing industry, there is an emerging mode of mass customization, like the production and operation mode of e-commerce. In this mode, the workshop directly contacts customers via e-commerce platforms and the Internet to design and manufacture customized products. In this way, the customized products can be designed and manufactured collaboratively in large quantities yet at a fast speed and a low cost. In this context, it is important to intellectualize the processing and manufacturing of products, especially workshop scheduling.

This paper mainly aims to design and implement an intelligent scheduling algorithm for an e-commerce workshop. With a large production scale, the workshop can manufacture products of various types and specifications, involving complex techniques, processes, and operations. During the production, every machine is available for processing the jobs. Thus, the scheduling of the workshop is a complex flexible workshop scheduling problem (FWSP). Based on the situation of e-commerce workshop and FWSP algorithms, genetic algorithm (GA) was integrated with deep learning neural network (DLNN) to solve the FWSP.

2. LITERATURE REVIEW

The WSP [1], which is extended from the two-machine ordered flow-shop scheduling problem (FSP), has been proved to be non-deterministic polynomial-time (NP) hard [2]. Through Lagrangian relaxation, Hoitomt et al. [3] theoretically solved the WSP by decomposing the

problem into multiple sub-problems. Brezovec et al. [4] improved the tabu search algorithm to handle the WSP.

In recent years, the AI algorithms have been introduced to solve the WSP. Moslehi and Mahnam [5] modelled the multi-objective FWSP with parallel machines, and presented a simple Pareto-based optimization algorithm. Dalfard and Mohammadi [6] created a hybrid GA to overcome the premature convergence, slow search, and poor solution of the WSP. Using the standard dataset, Shen and Yao [7] proved and solved the FWSP with small batches of various products, and successfully optimized the makespan, machine load, total load, etc. Based on earliest due date (EDD) and longest processing time (LPT), Mirabi [8] proposed a coding method to solve the unexpected situations that arise when the WSP is solved by the GA.

Since the conception of the FWSP, intelligent algorithms have achieved superior results on large scheduling problems. There are mainly three kinds of intelligent algorithms: evolutionary algorithms, swarm intelligence algorithms, and neural network algorithms.

(1) Evolutionary algorithms

The GA is a very common evolutionary algorithm with strong adaptability. The improved versions of this algorithm are widely applied to solve the FWSP. Chen et al. [9] developed a grouping GA, which focuses on parallel machine and re-entrant process in the WSP. Starting with the search method of the initial population, Wang et al. [10] integrated various search methods to improve the quality of the solution to the initial population, thereby accelerating the convergence of the GA. To improve the quality of global solution, Yang et al. [11] designed a coding method that preserves excellent genes at a high probability.

(2) Swarm intelligence algorithms

Swarm intelligence algorithms, inspired by the behaviours of the swarm of insects, are capable of solving the complex problem of distributed scheduling. Representative swarm intelligence algorithms include particle swarm optimization (PSO), ant colony optimization (ACO), and artificial bee colony algorithm (ABC). Wu et al. [12] proposed an ACO to overcome the time dependence in the FWSP. Thammano and Phu-ang [13] improved the ABC to optimize the completion time of the FWSP.

(3) Neural network algorithms

Fnaiech et al. [14] were the first to apply the neural network to the WSP: the Hopfield neural network was implemented for joint maintenance of the production process. Teymourifar and Ozturk [15] solved the FWSP by combining the neural network with the GA. Absorbing the merits of neural network and intelligent algorithms, Li et al [16] created a WSP algorithm that far exceeds the traditional algorithms in efficiency.

3. FWSP MODELLING

The FWSP [17] is an NP-hard problem more complex than the traditional WSP. There are mainly two sub-problems in the FWSP: operation sequencing and machine selection. The scheduling aims to minimize the makespan or processing cost by determining the processing sequence of each operation, and the start time of each operation on each machine. The FWSP can be formally defined as follows.

Suppose there are n jobs and m machines in the flexible workshop. Let $W_j, j = (1, 2, \dots, n)$ be the j^{th} job; n_j be the number of operations in the j^{th} job, which should be implemented in sequence; o_{ij} be the i^{th} operation of the j^{th} job, which can be implemented on one or more machines; M be the set of machines; M_k be the k^{th} machine; M_{ij} be the set of machines for o_{ij} ; t_{ijk} be the processing time of o_{ij} on M_k .

Then, the flexible workshop scheduling must satisfy the following constraints:

(1) For the continuity of processing, the processing cannot stop halfway, in addition to eliminating machine faults.

(2) To prevent the confusion of machines, each machine can only process one job intelligently at a time; each job can only be processed on one machine at a time.

(3) The operations of the same job have a sequence constraint, while those of different jobs do not.

(4) The installation time of a job is included in the processing time of each operation.

Our scheduling method fully considers the actual situation of e-commerce workshop. For convenience, the following symbols were defined:

(1) $M = \{1, 2, \dots, m\}$: the set of machines, where m is the number of machines;

(2) $N = \{1, 2, \dots, n\}$: the set of jobs, where n is the number of jobs;

(3) o_{ij} : the i^{th} operation of the j^{th} job;

(4) k : the serial number of the machine selected to process operation o_{ij} ;

(5) t_{ijk} : the processing time of operation o_{ij} on the k^{th} machine;

(6) st_{ij} : the start time of operation o_{ij} ;

(7) ct_{ij} : the completion time of operation o_{ij} ;

(8) d_i : the delivery date of job i ;

(9) c_i : the completion time of job i ;

(10) F_{ijk} : the processing cost of operation o_{ij} on the k^{th} machine per unit time;

(11) C_{max} : the maximum completion time (i.e., makepan) of a job;

(12) x_{ijk} : whether the k^{th} machine is selected to process operation o_{ij} .

According to the actual situation of the workshop, four objectives were designed for flexible workshop scheduling:

(1) Minimizing the makespan

The minimum makespan is the best indicator of the processing efficiency of the workshop, and also the most fundamental metric of the WSP. The makespan refers to the maximum completion time among all machines, i.e., the completion time of the job being processed with the longest time from the first to the last operations. The minimal makespan can be defined as:

$$f_1 = \min(\max_{1 \leq j \leq n} c_j) \quad (1)$$

(2) Minimizing the maximum machine load

In the actual production process, different machines are often selected to process a job at different loads. If the machines are selected improperly, the resource allocation might be unreasonable. To minimize the machine load and make better use of machines, the maximum machine load was chosen as a scheduling objective. The minimal maximum machine load can be defined as:

$$f_2 = \min(\max_{1 \leq j \leq n} \sum_{k=1}^{k_j} t_{ijk} \cdot x_{ijk}) \quad (2)$$

(3) Minimizing the total machine load

In the actual WSP, it is necessary to minimize both the minimum load of each machine, and the total load of all machines, such as to meet the other scheduling objectives. Thus, the total load of all machines was taken as a scheduling objective. The minimal total machine load can be defined as:

$$f_3 = \min \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^{k_j} t_{ijk} \cdot x_{ijk} \quad (3)$$

(4) Minimizing the total processing cost

The processing cost is the machine cost per unit time. The sum of all processing costs is called the total processing cost. To increase production profit, this paper chooses the minimal total processing cost as a scheduling objective:

$$f_4 = \min(\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^{k_j} t_{ijk} \cdot x_{ijk} \cdot F_{ijk}) \quad (4)$$

The multi-objective flexible workshop scheduling model needs to subject to the following constraints:

$$\begin{aligned}
 st_{ij} + x_{ijk} \times t_{ijk} &\leq ct_{ij} \\
 ct_{ij} &\leq st_{i(j+1)} \\
 ct_{ij} + t_{ijk} &\leq st_{ij} + A(1 - x_{ijk}) \\
 ct_{ij} &\leq st_{i(j+1)} + A(1 - x_{ijk}) \\
 st_{ij} &\geq 0, ct_{ij} \geq 0
 \end{aligned}
 \tag{5}$$

where, A is a sufficiently large positive number.

4. FWSP ALGORITHM BASED ON IMPROVED GA

4.1 Traditional GA in WSP

As shown in Fig. 1, the FWSP can be solved by the GA in five steps: encoding and decoding, population initialization, selecting fitness function, choosing genetic operators, and determines the termination conditions.

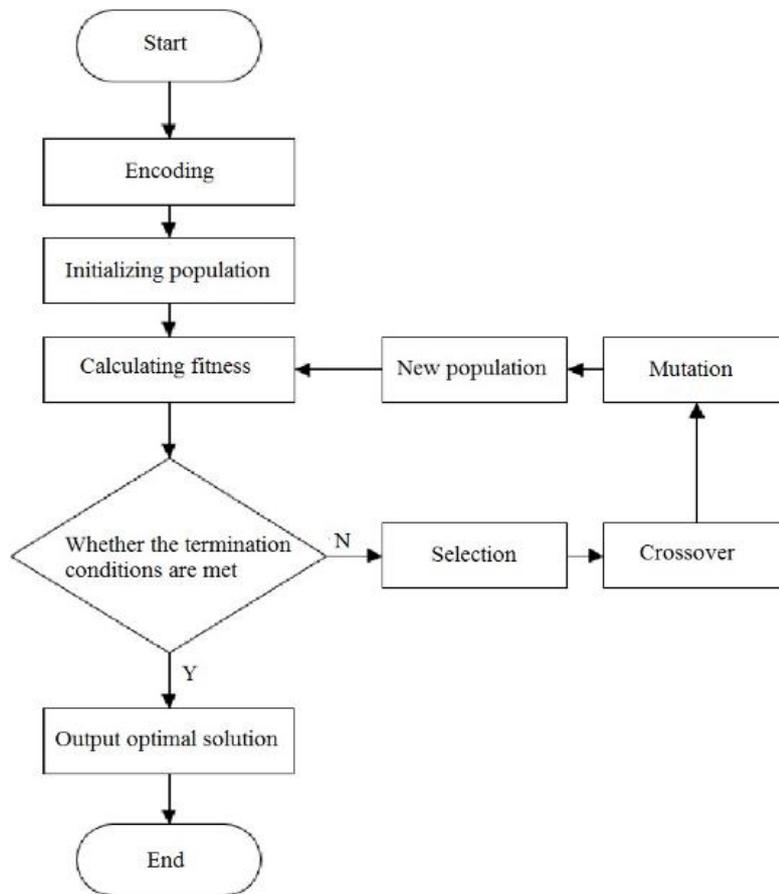


Figure 1: Flow chart of solving FWSP with traditional GA.

When the WSP is solved by the GA, the step of coding and decoding affects the transform between solution and chromosome, and plays a critical role in solving the entire problem. In the traditional WSP, the coding depends on operation sequencing. In contrast, the FWSP considers both operation sequencing and machine selection. Therefore, the coding method of the GA for the FWSP needs to combine operation-based coding with machine-based coding. Fig. 2 shows the chromosome combining the two coding methods.

The decoding is the reverse process of chromosome coding. Three decoding methods, namely, semi-active decoding, active decoding, and full-active decoding, can be selected according to actual needs. In this paper, the semi-active decoding is selected.

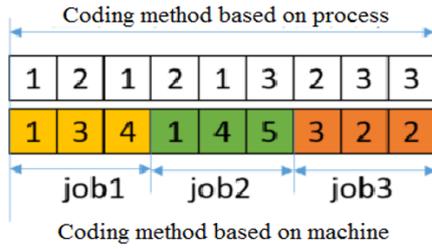


Figure 2: An example of chromosome combining the two coding methods.

As for the population initialization of the GA, the size of the initial population is smaller than the number of solutions in the solution space. However, the traditional method for population initialization does not have a good ability of global search to traverse the entire solution space in an evenly manner. Thus, this paper selects the Hamming distance to generate an evenly distributed initial population.

The selection of fitness function is an important step of the GA. Whether the fitness function is suitable directly bears on the convergence speed of the GA, and the possibility of iteratively findings the optimal solution. Considering the actual product demand of e-commerce workshop and the construction of fitness function in multi-objective FWSP, a weighted multi-objective fitness function can be designed as:

$$f_{fit} = 1/\beta_1\delta_1f_1 + \beta_2\delta_2f_2 + \beta_3\delta_3f_3 + \beta_4\delta_4f_4 \quad (6)$$

where, f_1 , f_2 , f_3 and f_4 are the four objectives of flexible workshop scheduling; β_i is the weight depending on the importance of each objective ($\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$); δ is a parameter that adjusts the fitness function according to the type of data in the application scenario, making it easy to compute the fitness.

In nature, fitness measures the ability of individuals in the population to adapt to the environment. A high fitness means a higher survival probability. In the GA, the individuals with a high fitness are retained in the next-generation population. Thus, it is crucial to select the individuals with a high fitness. Here, tournament selection and elite preservation are combined for the selection operation of the GA.

After the selection, crossover is needed to exchange the genes on parent chromosomes. According to the features of the FWSP, this paper chooses the floating-point Laplacian crossover operator, which enhances the survival probability of excellent individuals, enlarges the search space of the solution space, and improves the algorithm performance.

Next, mutation is implemented to generate new individuals by randomly mutating genes in chromosomes before joining the new population. The new individuals are added to the new population, making the range of solution larger, and the local search more effective. In this paper, the mutation operation is designed based on the operation- and machine-based coding method.

(1) Mutation derived from operation-based coding

Insertion mutation was designed from the operation-based coding method: the genes in chromosomes are randomly selected and inserted to random positions.

(2) Mutation derived from machine-based coding

In the FWSP, each operation can be implemented on one of multiple machines. During the processing, two optional machines with random operations were selected, the machine with the shortest processing time was chosen, and the machine number was coded into the chromosome to obtain the feasible solution.

4.2 Improved GA for WSP

The data prediction model for the WSP deals with the data on machines, jobs, completion times, machine loads, and energy consumption. In terms of prediction, deep learning does a better job

than the traditional machine learning. Inspired by deep learning, this paper establishes a DLNN called long short-term memory network (LSTM) [18-20] to predict the completion time, machine load, and energy consumption of some jobs in the manufacturing process. Integrating the features of industrial processing data with the superiority of the LSTM, the proposed model aims to forecast the results of the WSP quickly, so that the scheduling plan can be prepared in advance, making the scheduling more reasonable and orderly. Fig. 3 shows the structure of our model [21, 22].

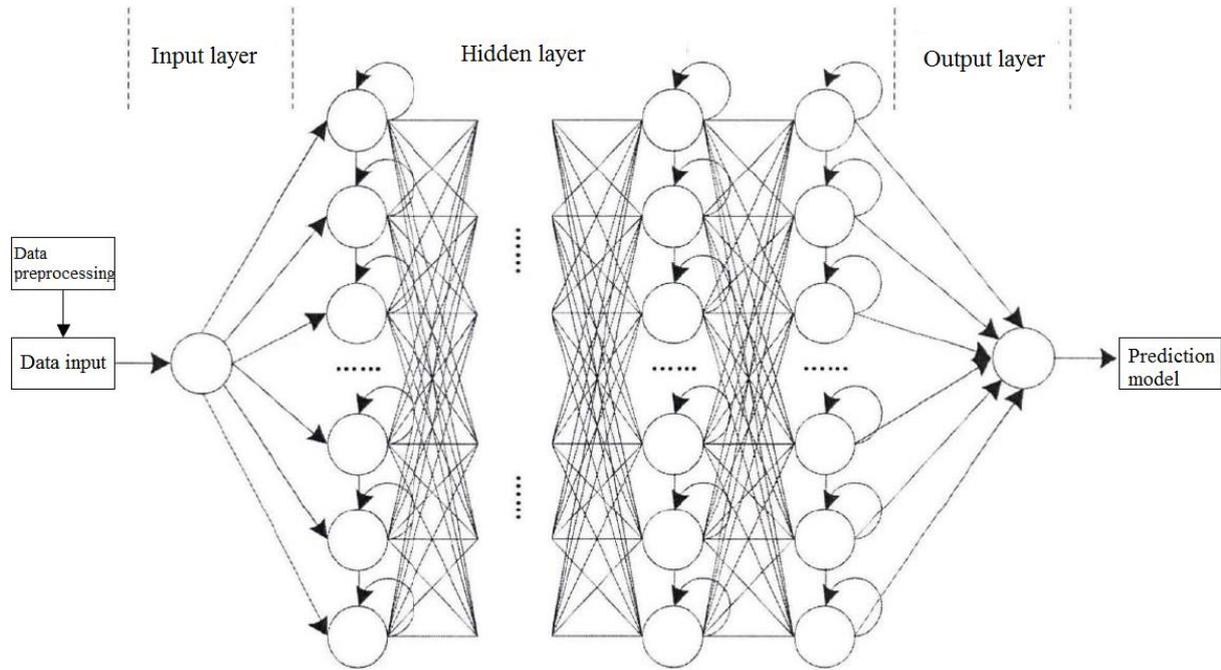


Figure 3: Structure of LSTM-based data prediction model.

From June to December 2020, the production data were collected from the production line of an e-commerce workshop, reflecting the variation in historical production operations. The LSTM is particularly good at processing such a time series. This ensures that the LSTM-based model can fit and extract the data features, and achieve a high prediction accuracy.

The effectiveness of the data prediction model hinges on the quality of the loss function, which serves as the optimization objective function in the DLNN. This paper adopts the mean squared error (*MSE*) as the loss function:

$$Lf(\theta) = 1/n \sum_{i=1}^n (y_i - f(x_i))^2 \quad (7)$$

where, $Lf(\theta)$ is the difference between the predicted value and the actual value; $f(x_i)$ is the fitted function; n is the number of training samples. During model training, the weights of each neuron were randomly generated at the beginning, and then updated through iterative training.

The data features are extracted by the hidden layer(s). In fact, the number of hidden layers determines the effect of feature extraction, and the prediction accuracy of the LSTM-based model. Here, continuous tests are conducted to identify the most suitable number of hidden layers. Specifically, the number of training cycles was set to 1,000, and the initial number of hidden layers was set to 1. In each test, the number of hidden layers was increased by one. Table I shows the prediction results at different number of hidden layers.

Table I: Prediction results at different number of hidden layers.

Number of hidden layers	1	2	3	4	5	6	7	8	9
<i>MSE</i>	21.91	20.66	20.95	20.61	20.63	20.71	21.03	21.08	21.11

As shown in Table I, the *MSE* of the data prediction model was minimized, in the case of four hidden layers. The prediction accuracy is negatively correlated with the value of the loss function. Therefore, our model had the highest accuracy, when there were four hidden layers.

Once the number of hidden layers was determined, the authors started to optimize the other hyperparameter that greatly affects the prediction effect: the number of hidden layer neurons. Similarly, multiple tests were conducted at different number of hidden layer neurons. The test results are shown in Table II.

Table II: Prediction results at different number of hidden layer neurons.

Number of hidden layer neurons	5	10	15	20	25	30	35
<i>MSE</i>	22.26	21.27	20.86	20.93	20.91	20.88	21.01

As shown in Table II, the *MSE* of the data prediction model was minimum, when each hidden layer had 15 neurons. To sum up, the proposed model had the best prediction effect, at four hidden layers, each of which contains 15 neurons.

For the FWSP, the GA has an outstanding global search ability, yet a weak local search ability. As a result, the convergence speed of the algorithm tends to slow down in the later phase. To solve the problem, the strengths of LSTM were integrated to stabilize the prediction model. After being trained, the data prediction model could obtain the stable mapping between the input data and output data on workshop scheduling, and extract and output the weight coefficient of each objective function. These coefficients were taken as the weights of the objective functions of the GA, thereby controlling the impact of each objective function on the scheduling results to a reasonable degree. This enhances the local search ability of the GA and speeds up its convergence.

In addition, the fitness function was improved in three steps: First, the weighted fitness function was obtained empirically; Second, the weight coefficient of each objective function was determined by the LSTM; Third, the weight coefficient was substituted to the fitness function to complete the improvement. Through the above steps, the authors proposed a fitness function of multi-objective GA combined with the LSTM:

$$f_{fit}(x) = 1/\beta_1\delta_1f_1(x) + \beta_2\delta_2f_2(x) + \beta_3\delta_3f_3(x) + \dots + \beta_n\delta_nf_n(x) \quad (8)$$

where, $f_i(x)$ is an objective function; β_i is the weight coefficient of the objective function, which is extracted by the LSTM.

In general, the proposed algorithm can be implemented in two stages: LSTM training, and GA operation:

Step 1. Analyse the WSP, initialize the algorithm parameters, and determine the coding method, objective functions, and relevant parameters of the WSP.

Step 2. Train the LSTM-based model.

Step 3. Extract and output the weight coefficient of each objective function by the LSTM.

Step 4. Initialize the population, and code the individuals in the solution space.

Step 5. Import the weight coefficients extracted by the LSTM to the fitness function.

Step 6. Compute the fitness of each individual in the population by the new fitness function.

Step 7. Perform selection, crossover, and mutation to generate a new population.

Step 8. If the termination conditions are met, output the optimal solution, i.e., the current maximum fitness; otherwise, repeat Steps 5 and 6 until the termination conditions are met.

5. SIMULATION AND RESULTS ANALYSIS

To verify its superiority, the proposed e-commerce workshop scheduling algorithm based on LSTM and GA was compared with the traditional GA. The simulation data (Table III) are about an FWSP with four jobs, each of which contains different operations; each operation could be

processed on one of four machines. Table IV compares the results of the two algorithms. It can be seen that our algorithm achieved better scheduling results than the traditional GA.

Table III: Simulation data.

Job	Operation	Processing time of optional machine				Processing cost
		M_1	M_2	M_3	M_4	
J_1	O_{11}	5	4	3	5	97
	O_{12}	3	4	-	4	95
	O_{13}	7	-	4	5	96
J_2	O_{21}	6	7	-	5	77
	O_{22}	3	6	8	-	79
	O_{23}	-	6	7	7	81
J_3	O_{31}	6	4	5	2	110
	O_{32}	2	6	4	3	121
	O_{33}	5	6	7	-	115
J_4	O_{41}	3	3	7	3	99
	O_{42}	4	6	7	4	96
	O_{43}	7	3	4	6	92

Table IV: Comparison of the results between the two algorithms.

Objectives	Traditional GA	Our algorithm
Makespan	36	31
Processing cost	9,964	5,743
Maximum machine load	156	112
Total machine load	187	163

6. CONCLUSIONS

Focusing on the FWSP, this paper designs a scheduling method suitable for solving various FWSPs, and proposes a multi-objective FWSP method based on the GA. According to the actual production data, a data prediction model was constructed from the DLNN called LSTM. Then, the GA was optimized by the LSTM to solve the multi-objective FWSP. Simulation results show that the proposed algorithm outshines the traditional GA in scheduling the e-commerce workshop, because it fully integrates the merits of LSTM and GA.

ACKNOWLEDGEMENT

This paper was supported by Anhui Social science planning general projects, Research on inter-regional co-construction and Performance of entrepreneurial ecosystem from the perspective of integrated development, AHSKY2020D09; Anhui Virtual Simulation Experimental Teaching Project, Data driven virtual simulation experiment teaching project of scientific product portfolio decision in high-tech manufacturing enterprises, 2019xfxm44.

REFERENCES

- [1] Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included, *Naval Research Logistics Quarterly*, Vol. 1, No. 1, 61-68, doi:[10.1002/nav.3800010110](https://doi.org/10.1002/nav.3800010110)
- [2] Garey, M. R.; Sethi, J. 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)
- [3] Hoitomt, D. J.; Luh, P. B.; Patipati, K. R. (1993). A practical approach to job-shop scheduling problems, *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 1, 1-13, doi:[10.1109/70.210791](https://doi.org/10.1109/70.210791)
- [4] Brezovec, C.; Cornuejols, G.; Glover, F. (1986). Two algorithms for weighted matroid intersection, *Mathematical Programming*, Vol. 36, No. 1, 39-53, doi:[10.1007/BF02591988](https://doi.org/10.1007/BF02591988)

- [5] 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)
- [6] Dalfard, V. M.; Mohammadi, G. (2012). Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints, *Computers & Mathematics with Applications*, Vol. 64, No. 6, 2111-2117, doi:[10.1016/j.camwa.2012.04.007](https://doi.org/10.1016/j.camwa.2012.04.007)
- [7] Shen, X.-N.; Yao, X. (2015). Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems, *Information Sciences*, Vol. 298, 198-224, doi:[10.1016/j.ins.2014.11.036](https://doi.org/10.1016/j.ins.2014.11.036)
- [8] Mirabi, M. (2014). A novel hybrid genetic algorithm to solve the sequence-dependent permutation flow-shop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, Vol. 71, No. 1, 429-437, doi:[10.1007/s00170-013-5489-5](https://doi.org/10.1007/s00170-013-5489-5)
- [9] Chen, J. C.; Wu, C.-C.; Chen, C.-W.; Chen, K.-H. (2012). Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm, *Expert Systems with Applications*, Vol. 39, No. 11, 10016-10021, doi:[10.1016/j.eswa.2012.01.211](https://doi.org/10.1016/j.eswa.2012.01.211)
- [10] Wang, X. J.; Liang, G.; 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)
- [11] Yang, J. J.; Ju, L. Y.; Liu, B. Y. (2011). The improved genetic algorithm for multi-objective flexible job shop scheduling problem, *Applied Mechanics and Materials*, Vol. 66-68, 870-875, doi:[10.4028/www.scientific.net/AMM.66-68.870](https://doi.org/10.4028/www.scientific.net/AMM.66-68.870)
- [12] Wu, J.; Wu, G. D.; Wang, J. J. (2017). Flexible job-shop scheduling problem based on hybrid ACO algorithm, *International Journal of Simulation Modelling*, Vol. 16, No. 3, 497-505, doi:[10.2507/IJSIMM16\(3\)CO11](https://doi.org/10.2507/IJSIMM16(3)CO11)
- [13] Thammano, A.; Phu-ang, A. (2013). A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem, *Procedia Computer Science*, Vol. 20, 96-101, doi:[10.1016/j.procs.2013.09.245](https://doi.org/10.1016/j.procs.2013.09.245)
- [14] Fnaiech, N.; Fitouri, C.; Varnier, C.; Fnaiech, F.; Zerhouni, N. (2015). A new heuristic method for solving joint job shop scheduling of production and maintenance, *IFAC-PapersOnLine*, Vol. 48, No. 3, 1802-1808, doi:[10.1016/j.ifacol.2015.06.348](https://doi.org/10.1016/j.ifacol.2015.06.348)
- [15] Teymourifar, A.; Ozturk, G. (2018). A neural network-based hybrid method to generate feasible neighbors for flexible job shop scheduling problem, *Universal Journal of Applied Mathematics*, Vol. 6, No. 1, 1-16, doi:[10.13189/ujam.2018.060101](https://doi.org/10.13189/ujam.2018.060101)
- [16] Li, Z. C.; Qian, B.; Hu, R.; Chang, L. L.; Yang, J. B. (2019). An elitist nondominated sorting hybrid algorithm for multi-objective flexible job-shop scheduling problem with sequence-dependent setups, *Knowledge-Based Systems*, Vol. 173, 83-112, doi:[10.1016/j.knosys.2019.02.027](https://doi.org/10.1016/j.knosys.2019.02.027)
- [17] Bin, S.; Sun, G. (2020). Optimal energy resources allocation method of wireless sensor networks for intelligent railway systems, *Sensors*, Vol. 20, No. 2, Paper 482, 20 pages, doi:[10.3390/s20020482](https://doi.org/10.3390/s20020482)
- [18] Sun, G. X.; Bin, S.; Jiang, M.; Cao, N.; Zheng, Z.; Zhao, H.; Xu, L. (2019). Research on public opinion propagation model in social network based on blockchain, *Computers, Materials & Continua*, Vol. 60, No. 3, 1015-1027, doi:[10.32604/cmc.2019.05644](https://doi.org/10.32604/cmc.2019.05644)
- [19] Wang, L. P. (2020). An improved long short-term memory neural network for macroeconomic forecast, *Revue d'Intelligence Artificielle*, Vol. 34, No. 5, 577-584, doi:[10.18280/ria.340507](https://doi.org/10.18280/ria.340507)
- [20] Terai, A.; Chiba, T.; Shintani, H.; Kojima, S.; Abe, S.; Koshijima, I. (2020). Intrusion detection using long short-term memory model for industrial control system, *International Journal of Safety and Security Engineering*, Vol. 10, No. 2, 183-189, doi:[10.18280/ijss.100204](https://doi.org/10.18280/ijss.100204)
- [21] Ojstersek, R.; Acko, B.; Buchmeister, B. (2020). Simulation study of a flexible manufacturing system regarding sustainability, *International Journal of Simulation Modelling*, Vol. 19, No. 1, 65-76, doi:[10.2507/IJSIMM19-1-502](https://doi.org/10.2507/IJSIMM19-1-502)
- [22] Vrecko, I.; Kovac, J.; Rupnik, B.; Gajsek, B. (2019). Using queuing simulation model in production process innovations, *International Journal of Simulation Modelling*, Vol. 18, No. 1, 47-58, doi:[10.2507/IJSIMM18\(1\)458](https://doi.org/10.2507/IJSIMM18(1)458)