NSGA-II algorithm and application for multi-objective flexible workshop scheduling

Wang Yahui ⓘ**, Shi Ling, Zhang Cai, Fu Liuqiang and
Jin Xiangjie**

Abstract
Based on the study of multi-objective flexible workshop scheduling problem and the learning of traditional genetic algorithm, a non-dominated sorting genetic algorithm is proposed to solve and optimize the scheduling model with the objective functions of processing cycle, advance/delay penalty and processing cost. In the process of optimization, non-dominated fast ranking operator and competition operator are used to select the descendant operator, which improves the computational efficiency and optimization ability of the algorithm. Non-repetitive non-dominant solutions and frontier sets are found in the iteration operation to retain the optimal results. Finally, taking a manufacturing workshop as an example, the practicability of the proposed algorithm is verified by the simulation operation of the workshop scheduling information and the comparison with other algorithms. The results show that the algorithm can obtain the optimal solution more quickly than the unimproved algorithm. The improved algorithm is faster and more effective in searching, and has certain feasibility in solving the job shop scheduling problem, which is more suitable for industrial processing and production.

Keywords
Multi-objective, workshop scheduling, non-dominated sorting, genetic algorithm

Received 21 November 2019; revised 16 April 2020; accepted 22 June 2020

Introduction

With the development of science and technology and the application and popularization of advanced manufacturing technology, manufacturing companies are also facing the pressure of survival such as labour costs, equipment replacement and the renewal of advanced manufacturing concepts. Many manufacturing enterprises are committed to improving energy and material efficiencies. To achieve this goal, one of the high-efficiency ways is to optimize the scheduling.[1,2] Facing with the diversification of market demand and the renewal of processing technology, the production and manufacturing models has also begun to change towards convenience and flexibility. In real-world manufacturing systems, production scheduling systems are often implemented under random or dynamic events like machine failure, unexpected processing times, stochastic arrival of the urgent orders, cancellation of the orders and so on. These dynamic events will lead the initial scheduling scheme to be nonoptimal and/or infeasible. Hence, appropriate dynamic rescheduling approaches are needed to overcome the dynamic events. Therefore, the quality of the scheduling scheme is of great significance to the company's revenue and to cope with the changing market environment.[3,4]

Multi-objective flexible workshop scheduling is an extension of single-objective scheduling.[5] In the flexible workshop scheduling problem, each process in each component is not only fixed on one machine, different machines can be selected for processing according to

School of Machinery, North China University of Water Resources and Electric Power, Zhengzhou, China

Corresponding author:
Wang Yahui, School of Machinery, North China University of Water Resources and Electric Power, Zhengzhou 450045, China.
Email: wangyahui6@yeah.net

the specific circumstances, and the processing time of different machines is different, which is more flexible than traditional workshop scheduling.[6]

Description and model construction of multi-objective flexible workshop scheduling problem

Description of the problem

Assuming that K machines in a workshop plan to produce N workpieces, each workpiece needs more than one process M processing and the processing sequence can only be arranged in sequence, also the machine equipment is intelligent equipment which can process many processes. The process can select machinery and equipment according to production needs, and the machine can only occupy once at the same time, besides that the workpiece can only be processed by one machine at the same time. The processing time depends on the performance of the selected equipment and the complexity of the process.[7] Multi-objective flexible job-shop scheduling involves mainly parallel processing and multi-function machines.[8] The machine can process multiple processes without mutual interference. The machine has high adaptability and versatility, and can be interchanged according to the needs in the process of processing. For example, the process of the set workpiece is processed on the machine, and the machine is processing other parts in an occupied state within a predetermined time. At this time, the replacement machine can also meet the processing needs.[9]

A two-dimensional table is used to describe the flexible scheduling problem as shown in Table 1. The information corresponding to the values in the table is that the process of the workpiece can be processed on the machine. '•' means that the processing conditions are not satisfied. If V_{yz} indicates that the machine K_x does not meet the process M_z conditions of the workpiece N_y. For example, Vyz indicates that Kx does not meet the process Mz of the workpiece Ny. The implementable set of numbers for processing and production is expressed as KF_{yz}, $1 \leq x < K$, $1 \leq y < N'$ $1 \leq z < M$. The process M_z in which the workpiece N_y is represented by U_{yz} can select the set numbers of feasible solutions of the machining on the machine as $U_{yz} \subseteq$

$\{1, 2 \cdots K\}$ T_{xyz} is the processing time of process M_z of workpiece N_y on machine K_x.

In order to simplify the problem model, it is assumed that the machine is in the state to be processed at time $t = 0$ and the processing objects are executed according to the original plan and processed sequentially, then the flexible job-shop scheduling problem (FJSP) can obtain the following problems from the above expression.

Fully flexible FJSP: For $V_{yz}, |KF_{yz}| = x$, the process of the workpiece is not restricted by the machine, and the machine in the non-occupied state can satisfy its working conditions, such as the process m_{11} in the workpiece N_1, the process m_{21} in the workpiece N_2, and the process m_{31} and process m_{34} in the workpiece N_3.

Partial flexible FJSP: for $V_{yz}, |KF_{yz}| < x$, the working procedure is restricted by the machining conditions. Even if the machine is idle in the arrangement, the machine can only wait or reschedule because it does not meet the use performance, such as the process m_{12} in the workpiece N_1, the process m_{22} in the workpiece N_2, and the process m_{32} and process m_{33} in the workpiece N_3.

Optimized mathematical model construction

Problem model construction.

1. The objective function is established by taking the minimum period of the workpiece flow during processing as a measure

$$\min f_1 = \sum_{0=1}^{0} \sum_{m=1}^{M} \sum_{m=1}^{N} (T_{omn} - t_{omn}) \tag{1}$$

2. Based on the delivery date and the punctuality of product production plan according to schedule is used to measure the delay completion penalty or the early completion penalty to establish the objective function

$$\min f_2 = \sum_{m=1}^{M} r_{m1} * \max(0, K_{omn} - d_m) \tag{2}$$

Table 1. Two-dimensional description of flexible FJSP.

Workpiece	Process	Machine K_1	Machine K_2	Machine K_3	Machine K_4
N_1	m_{11}	1	2	5	4
	m_{12}	•	1	3	1
N_2	m_{21}	2	2	4	1
	m_{22}	1	•	2	•

$$\min f_3 = \sum_{m=1}^{M} r_{m2} * \max(0, K_{\text{omn}} - d_{\text{m}}) \qquad (3)$$

3. The objective function is established by taking the minimum processing cost as the measurement value

$$\min f_4 = \sum_{0=1}^{0} \sum_{m=1}^{M} \sum_{m=1}^{N} U_{\text{omn}} + f_2 + f_3 \qquad (4)$$

where 'm' is the serial number of the workpiece, $m = 1,2,3, \ldots, m$, 'n' is process number, $n = 1,2,3,\ldots,n$, 'o' is the serial number of the production machine, k_{omn} indicates that the process n of the workpiece m is processed on the machine o. U_{omn} indicates the production cost of the process n of the workpiece m when it is processed on the machine o. t_{omn} indicates the starting time when the workpiece m's process n is machined on the machine o. The processing time when the process n is completed is indicated by t_n. The processing time of the workpiece m is indicated on the machine o by T_{omn}. The workpiece m delivery date is indicated by d_m. The penalty factor for the completion of the workpiece m delay is indicated by . The penalty factor for the pre-completion of the workpiece m is indicated by r_{m2}.

Constraints.

$$\begin{cases} k_{omn} - k_{\text{o(m-1)n}} - \text{o}_{omn} \geq 0 \\ x_{omn} = \ \text{x}_{\text{o(m-1)n}} \\ 1 < m \leq M \end{cases} \qquad (5)$$

$$\begin{cases} k_{omn} - k_{oij} - o_{omn} \geq 0 \\ x_{omn} = x_{oij} = 1 \\ t_{omn} > t_{oij} \end{cases} \qquad (6)$$

When the workpiece is machined on the machine, the next step of the workpiece can be carried out after the process of the workpiece on the same machine is completed

$$\begin{cases} T_{o(m+1)n} \geq t_{omn} + T_{omn} \\ T_{omn} \geq t_{omn} \\ t_n > 0 \end{cases} \qquad (7)$$

After the completion of the technological process of the workpiece, the processing time is sequential and the processing time of the same process is fixed, which involves the waiting time of the machine stopping. Therefore, the processing time of the workpiece on the machine is longer than that on the machine, and the processing time of the same process is fixed. The working time should not be zero in time setting.

NSGA-II algorithm operation flow

For the multi-objective coexistence of workshop scheduling, the traditional standard genetic algorithm cannot meet the needs of multi-objective constraints. Non-dominated sorting genetic algorithm (NSGA) is proposed based on the needs of multi-objective constraints. NSGA is based on the theory of algorithm, which is the same as the recombination operator mechanism of the standard algorithm, except that the selection operation factor is different. It only uses the non-dominated sorting stratification and fitness value calculation for the population, and uses the niche technology to determine the fitness to ensure the distribution characteristics of the Pareto optimal solution set,[10,11] and then performs algorithm operations. However, the computational complexity of NSGA is high and the non-deterministic elite retention mechanism is limited in use. Based on its advantages, Srinivas and Deb proposed the NSGA-II algorithm. The NSGA-II algorithm makes up for the shortcomings of NSGA in use and reduces the computational complexity. Instead of sharing radius share Q, the comparison operator of congestion degree and congestion degree is used. The population diversity is guaranteed by extending the individuals in the quasi-Pareto domain to the whole Pareto domain with uniform distribution. The elite strategy adopted also improves the speed and stability of the algorithm.[12,13]

When using NSGA-II, the chromosomes in the initialized population are sorted by non-dominated sorting and are only constrained by the first dominant frontier. The next order of the chromosomes is constrained by the first order, and each step has a ladder value assigned by the chromosomes and is arranged sequentially. The chromosome also uses the crowded distance as a parameter to measure the distance from other chromosomes. The smaller the distance is, the more the individual is, the lower the diversity will be; among the multiple chromosomes under the same ladder, the chromosomes with smaller ladder value and larger crowded distance have a higher probability of generating offspring as the next round of operation of the father generation. The method of selecting the father generation is to adopt the binary Championship method.

Coding and decoding

In the coding of the NSGA-II algorithm for solving the workshop scheduling, the initial population of genetic operations in NSGA-II algorithm is formed by selecting the processing procedure 'n' of workpiece 'm' to process on machine 'o'. The chromosomal coding moments that can be performed are

$$O = \begin{bmatrix} n_{11} & n_{12} & \dots & n_{1N} \\ n_{21} & n_{22} & \dots & n_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ n_{M1} & n_{M2} & \dots & n_{MN} \end{bmatrix}$$

The corresponding in the matrix is the arrangement of $M * N$ machining plans on the machine 'o'. Each row in the matrix is a process. Each column represents a workpiece. If different workpieces are processed in the same process during processing, the processing time can be expressed as

$$T_{onx} = T_{on1} - t_{on1} + t_{n2} + \dots + t_{nx} \qquad (8)$$

That is, the time required to process x workpieces.

If it is the first process, the order of processing is $n_{11}, n_{12} \dots n_{1N}$, and if the workpiece is processed sequentially, the order of processing is $n_{11}, n_{21} \dots n_{M1}$. For non-identical workpieces and processes, in order to shorten the waiting time of the machine, the principle of near processing is adopted. In the following, the process and workpiece of 3×3 are processed in the workshop scheduling processing example. The machines that can be processed in the process are 2 2 3, respectively, and the obtained chromosome coding matrix is

$$O = \begin{bmatrix} 1.26 & 1.35 & 2.51 \\ 1.32 & 1.25 & 1.21 \\ 1.62 & 1.32 & 1.23 \end{bmatrix}$$

For example, the processing sequence in the matrix, if the second process of the workpiece 2 and 3 is simultaneously processed on the machine, since $1.25 > 1.32$, the workpiece 2 is processed first and then the workpiece 3 is processed when the processing sequence is selected; when the third process of the workpiece 1 and 3 is machined on the machine, since $1.26 < 2.51$, the workpiece 3 is preferentially machined in the selection process and then the workpiece 2 is processed. If the follow-up process is not started until the first process of each workpiece has been completed, since $1.26 < 1.36 < 2.51$, the sequence of the completion of the first process is the workpiece 3, the workpiece 2 and the workpiece 1. The processing of the same workpiece is constrained by the routing of the process.

Fast non-dominated sorting

In the algorithm, the individual of the population is divided into different levels according to the dominance relationship, and the different grades of n individuals

are represented by $MN_1, MN_2 \dots MN_n$, and the corresponding relationships are as follows

$$U_{MN \in \{MN_1, MN_2 \dots MN_n\}} MN = O_{POP} \qquad (9)$$

$$\forall_{i,j} \in \{1, 2 \dots n\}, i \neq j, MN_i \cap MN_j = \Phi \qquad (10)$$

$$MN_1 \succ MN_2 \succ \dots \succ MN_n, (k = 1, 2, \dots n) \qquad (11)$$

Equation (9) indicates that the union of individuals in each rank is the whole population; equation (10) indicates that there is only one non-dominated set of individuals; equation (11) indicates the dominant relationship among different population grades. The maximum number of accesses in the non-dominant ordering of non-first dominance i is $(N-1)$, and the number of non-dominated frontiers is a non-zero constant, that is $n \geq 1$.

Operator reorganization

Selecting an operation. The coding is random at the time of sorting, so the probability of the process being selected when the algorithm searches is the same. In order to improve the search efficiency of the algorithm, we usually evaluate the fitness function of the edited string to select the appropriate string for the next iteration. For example, if the string of the process n_1, n_2, n_3 is better than the n_1, n_3, n_2 when machining on the machine o, the n_1, n_2, n_3 value at the time of the operator operation is high. In multi-objective flexible workshop scheduling, usually one machine corresponds to multiple processing operations, that is

$$\begin{bmatrix} n_1 & n_2 & n_3 & o_2 & o_3 \\ n_2 & n_1 & n_3 & o_1 & o_4 \\ n_1 & n_3 & n_2 & o_2 & o_3 \end{bmatrix}$$

The first half shows the machining sequence of the workpiece 'n', and the latter shows the machining machine that the workpiece can select. [0, 1] indicates that the use conditions of the machine 'o' processing step 'n' are satisfied, '0' indicates that machine 'o' does not satisfy the working procedure 'n' at the current time, and '1' indicates that machine 'o' satisfies the working procedure 'n'. The corresponding constraint is

$$o_x = \begin{cases} 1 & k_{omn} = 1 \\ 0 & k_{omn} = 0 \end{cases} \qquad (12)$$

k_{omn} indicates that the process 'n' of the workpiece 'm' is processed on the machine 'o'.

Cross operation. Taking chromosome coding matrix for example

$$O_1 = \begin{bmatrix} 1.26 & 1.35 & 2.51 \\ 1.32 & 1.25 & 1.21 \\ 1.62 & 1.32 & 1.23 \end{bmatrix}$$

$$O_2 = \begin{bmatrix} 1.45 & 1.27 & 1.92 \\ 1.17 & 1.39 & 1.78 \\ 1.19 & 1.47 & 1.39 \end{bmatrix}$$

Selecting the row intersection and the intersection point is '1'. Getting the new matrix

$$O_1 = \begin{bmatrix} 1.26 & 1.35 & 2.51 \\ 1.17 & 1.39 & 1.78 \\ 1.62 & 1.32 & 1.23 \end{bmatrix}$$

$$O_2 = \begin{bmatrix} 1.45 & 1.27 & 1.92 \\ 1.32 & 1.25 & 1.21 \\ 1.19 & 1.47 & 1.39 \end{bmatrix}$$

The corresponding matrix represents the change of the machine usage corresponding to process A of the workpiece. Selecting column intersection and intersection point 2 to get a new matrix as follows

$$O_1 = \begin{bmatrix} 1.26 & 1.35 & 1.92 \\ 1.32 & 1.25 & 1.78 \\ 1.62 & 1.32 & 1.39 \end{bmatrix}$$

$$O_2 = \begin{bmatrix} 1.45 & 1.27 & 2.51 \\ 1.17 & 1.39 & 1.21 \\ 1.19 & 1.47 & 1.23 \end{bmatrix}$$

The corresponding matrix can be expressed as a change in the machine used for the machined workpiece.

Mutation operation. In order to preserve a better population, the two intersecting individuals with a non-dominated rank greater than 2 are calculated by Hamming distance. If the similarity is greater than the threshold value, the mutation rate parameter is increased, and vice versa. In the production scheduling, in order to improve the work efficiency, queue-insertion processing is carried out, that means the workpiece process satisfies the machine's queue insertion for the process of workpiece being processed, and the condition of the workpiece using the machine 'o' is

$$\begin{cases} T_{om_2n_2} - t_{om_2n_2} > T_{om_1n_1} - t_{om_1n_1} \\ T_{om_2n_2} - t_{om_2n_2} > t_{on_1} \end{cases} \qquad (13)$$

The change of the processing sequence code can be divided into the machine code changes and the process code changes, namely

$$\begin{bmatrix} n_1 & n_2 & n_3 & | & o_2 & o_3 & \dots & o_x \\ n_2 & n_1 & n_3 & | & o_1 & o_4 & \dots & \\ n_1 & n_3 & n_2 & | & o_2 & o_3 & \dots & \end{bmatrix} \text{ or}$$

$$\begin{bmatrix} n_1 & n_2 & n_3 \dots n_x & | & o_2 & o_3 \\ n_2 & n_1 & n_3 \dots & | & o_1 & o_4 \\ n_1 & n_3 & n_2 \dots & | & o_2 & o_3 \end{bmatrix}$$

If the corresponding process code of the machine is [1,3,5,2,3,4,2,2], the process code will be changed to [1,3,5,2,7,8,3,4,2,2] after inserting the process when the machine is not occupied.

Elite retention strategy

In the operation of non-dominant sorting algorithm, each generation will choose the better population of the parent to participate in the reorganization of the next generation population, retaining the non-dominated solution in the evolution process, and the speed of the algorithm searching for the optimal surface is improved.[14] Although most of the solutions in evolution will approach the non-dominated frontier surface, due to the different qualities of the surface, the surface solution set that may be approximated is the surface of the non-problem optimal solution set. In order to make the algorithm solution set to the problem surface search, the elite strategy is shown in equation (14)

$$n_i = |Front_i| \times r_i \qquad (14)$$

where n_i represents the number of individuals selected on the ith non-dominated surface Front; $|Front_i|$ represents the number of individuals on the ith layer non-dominated surface; r_i represents a random number, $r_i \in (0, 1)$.

Fitness function

In the algorithm operation, the feasible conditions of the processing will perform an evaluation function to evaluate the fitness and keep the optimal.[15] Fitness evaluation of processable gene strings is carried out in operator recombination, such as the fitness evaluation function (15) used in the third chapter.

$$F_I = \exp\left(\frac{f_I}{t}\right) \qquad (15)$$

Table 2. Machine processing time corresponding to workpiece process.

Workpiece number	Process	M1	M2	M3	M4	M5	M6
1	1	*	20	16	18	*	13
	2	16	14	19	*	17	*
	3	19	*	22	19	25	21
	4	14	13	*	12	16	*
2	1	13	17	*	11	*	13
	2	17	20	*	16	*	18
	3	*	*	15	11	19	20
	4	17	*	21	*	14	21
3	1	*	19	*	17	19	22
	2	*	16	17	15	*	19
	3	16	15	*	19	*	18
	4	19	*	17	16	11	*
4	1	13	11	*	7	9	*
	2	*	23	28	29	*	24
	3	24	23	*	29	*	21
	4	25	*	26	22	21	*
5	1	*	22	23	30	*	36
	2	24	*	29	*	23	26
	3	26	21	23	24	*	28
	4	*	22	26	25	29	*
6	1	17	19	*	20	28	26
	2	19	*	15	13	*	19
	3	*	23	24	*	17	18
	4	23	28	*	27	*	21

Note: Time unit: minute.

Table 3. Machine-processing costs corresponding to workpiece processes

Workpiece number	Process	M1	M2	M3	M4	M5	M6
1	1	*	9	7	11	*	9
	2	9	7	8	*	11	*
	3	13	*	16	14	11	9
	4	11	9	*	12	15	*
2	1	12	17	*	13	*	11
	2	8	11	*	8	*	9
	3	*	*	11	10	14	15
	4	8	*	12	*	10	7
3	1	*	13	*	11	15	12
	2	*	12	13	10	*	9
	3	11	9	*	13	*	17
	4	13	*	12	11	8	*
4	1	11	13	*	9	10	*
	2	*	11	13	11	*	17
	3	14	12	*	9	*	17
	4	12	*	8	10	14	*
5	1	*	12	13	8	*	11
	2	16	*	20	*	13	17
	3	14	11	11	13	*	9
	4	*	11	17	14	13	*
6	1	13	10	*	9	12	14
	2	9	*	12	14	*	10
	3	*	14	11	*	16	17
	4	15	13	*	11	*	9

Note: Processing cost unit: yuan.

where '*t*' represents a change parameter, F_I represents a fitness function value, f_I represents an objective function value, and I represents a corresponding chromosome. The established objective function has an appropriate reference standard in the operation, and the ideal result is achieved in the experiment. In the calculation, the target values are weighted and the fitness of the individual is evaluated. For example, the matrix of the order of processing *n* for workpiece *m* is

$$x_1 = |1, 5, 3|, \ x_2 = |4, 1, 6|, \ x_3 = |2, 3, 1|$$

x_1, x_2, x_3 is weighted according to the evaluation function. If the fitness function is $x_1 = x_2 = x_3$ before the operation, the operation is assigned according to the processing order and the machine selection, w_1, w_2, w_3 and $w_1 > w_3 > w_2$, the assigned gene coding will generate the priority of the gene string entering the next operation. This is $x_1 > x_3 > x_2$.

Table 4. The workpiece process corresponds to the delivery period.

Workpiece number	Delivery date	Delay penalty	Punish in advance
1	115	0.5	0.2
2	115	0.4	0.2
3	110	0.6	0.1
4	110	0.6	0.1
5	120	0.5	0.3
6	120	0.5	0.2

Note: Delivery unit: minute; advance/penalty unit: yuan/min.
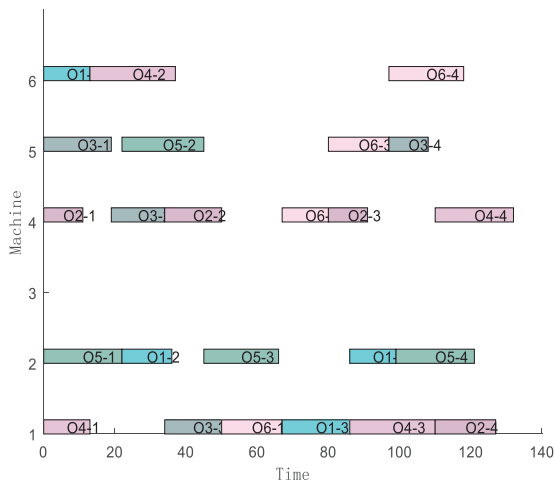
Example simulation analysis

In order to verify the validity of the algorithm, a $6 \times 6 \times 4$ optimization model is obtained after processing the data of a processing workshop production line of a bearing company. After the product information is collected, the arrangement is shown in Tables 2 to 4, '*' indicates that the corresponding machine does not meet the usage requirements, and the population size is 20; the number of iterations is 200; the mutation rate is 0.1 and the crossover rate is 0.7.

The optimized results obtained are shown in Figures 1 and 2. According to the scheduled Gantt chart and the processing information, the machine 3 is completely idle, that means the optimal scheduling result is achieved under the premise of using five machines, and it improves the use efficiency of other machines and reduces the no-load time of the machines. Because of the scheduling of four objectives, the Pareto frontier of the first three objectives is selected as shown in the following figure. There are 14 optimization solutions for the objectives. The program runs to get $f_1 = 414$, $f_2 = 18.5$, $f_4 = 436.3$. In order to verify the effectiveness of the algorithm, the Gantt chart obtained by the traditional genetic algorithm and the particle swarm algorithm is shown in Figure 3.

For the simultaneous optimization of the four targets, the comparison results of the three algorithms are shown in Table 5.

Through the comparison of the above three algorithms, the following conclusions can be drawn: The algorithm saves 58 and 7 min compared to traditional genetic algorithm and particle swarm optimization when optimizing objective function 1. When optimizing the objective function 2, it is reduced by 0.1 and 29.1 yuan, respectively, compared with the traditional genetic algorithm and the particle swarm algorithm; When optimizing the objective function 3, it saves 3
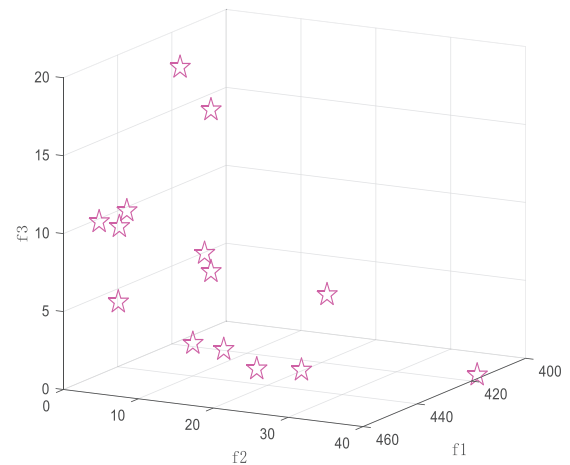

Figure 1. NSGA-II algorithm Gantt diagram.

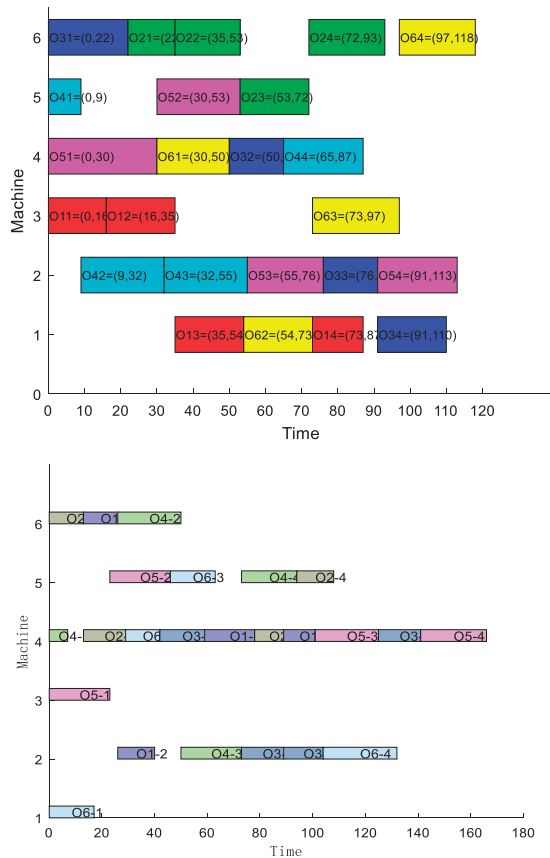**Figure 2.** NSGA-II algorithms Pareto frontier.

Figure 3. Gantt diagrams of genetic algorithm and particle swarm algorithm.

Table 5. Comparison of algorithm optimization results.

Algorithm	Target 1	Target 2	Target 3	Target 4
Traditional genetic algorithm	472	18.7	6.8	466.3
Particle swarm algorithm	421	47.6	5.8	474.4
NSGA-II algorithm	414	18.5	3.8	436.3

and 2 yuan, respectively, compared with the traditional genetic algorithm and the particle swarm algorithm; When optimizing the objective function 4, it is reduced by 30 and 38.1 yuan, respectively, compared with the traditional genetic algorithm and the particle swarm algorithm. The comparison shows that the algorithm has good performance. From the perspective of the scheduling Gantt chart distribution, the algorithm used can maximize the use of the machine and reduce the idle waiting time of the machine. Although it takes longer to complete the machining of the workpiece at the same

time than the traditional genetic algorithm, it can accomplish the processing objectives under the condition of using five machines in terms of machine occupancy.

Conclusion

Through the learning of multi-objective flexible workshop scheduling, we learned how to numerically process each production link and elements and used the learned workshop scheduling knowledge to establish a multi-objective workshop scheduling model as well as defined the symbols in the model; Through the study of the theoretical knowledge of genetic algorithm, the NSGA-II algorithm is extended, the application of the improved genetic algorithm in solving job shop scheduling and the comparison with the results of other algorithms prove that the improved genetic algorithm has good practical performance in solving multi-objective workshop scheduling problems.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ORCID iD

Yahui Wang https://orcid.org/0000-0002-4016-7871

References

1. Shaoqun X. Research on the technical status and development of workshop scheduling problem, North China University of Water Resourcesand Electric Power, 2019: 02; 1006–8554, 86–87.
2. Bhosale KC. Material flow optimisation of flexible manufacturing system using real coded genetic algorithm (RCGA). *J Mater Today Proc* 2008; 05: 7160–7167.
3. Renqingdaoerji. *Multi-objective models establishment and algorithm for job shop scheduling problem*. Xi'an: Xidian University, 2013, pp. 1–7.
4. Lei W, Jincao C and Dunbing T. Flexible job shop scheduling based on improved genetic algorithm. *J Nanjing Univ Acronaut Astronaut* 2017; 49: 779–785.
5. Wang Y, Fu L and Su Y. Genetic algorithm in flexible work shop scheduling based on multi-objective optimization. *J Interdiscip Math* 2018; 21: 1249–1254.
6. Rui C, Xiangpan H and Siting J. Research on flexible workshop scheduling problem based on improved genetic algorithm. *J Comp Digital Eng* 2019; 02: 285–288.

7. Xiuli W. *Research on multiobjective flexible job shop scheduling problem*. Xi'an: Northwestern Polytechnical University, 2006, pp. 1–10.

8. Xiong J, Xing L and Chen Y. Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. *Int J Prod Econ* 2013; 141: 112–126.

9. Smith WE. Various optimizers for single stage production. *Nav Res Logist Q* 1956; 3: 59–66.

10. Hyun CJ and Kim YK. A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Comp Oper Res* 1998; 7: 7–8.

11. Demir Y and İşleyen SK. Evaluation of mathematical models for flexible job-shop scheduling problems. *Appl Math Model* 2013; 37: 977–988.

12. Xi W, Qiao B and Zhu J. Flexible job shop scheduling based on improved genetic algorithm. *J Harbin Inst Technol* 2007; 39: 1151–1153.

13. Zhang C. Improved multi objective evolutionary algorithm for solving flexible job shop scheduling problem. *Jisuanji YU Xiandaihua*. 2017; 9: 80–84.

14. Ling W. *Shop scheduling with genetic algorithms*. Beijing: Tsinghua University Press, 2003, pp. 39–40.

15. Bergh F. *An analysis of particle swarm optimizers*. Pretoria: Department of Computer Science, University of Pretoria, South Africa, 2002, pp. 938–946.