Journal of Intelligent Management Decision
https://www.acadlore.com/journals/jimd

Integrated Multi-objective Optimization of Predictive Maintenance and Production Scheduling: Perspective from Lead Time Constraints

Zhiyuan Zhao, Qilong Yuan*

School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, 710048 Xi'an, China

* Correspondence: Qilong Yuan (ccyxz@xaut.edu.cn)

Received: 06-02-2022 **Revised:** 07-19-2022 **Accepted:** 08-23-2022

Citation: Z. Y. Zhao and Q. L. Yuan, "Integrated multi-objective optimization of predictive maintenance and production scheduling: Perspective from lead time constraints," *J. Intell Manag. Decis.*, vol. 1, no. 1, pp. 67-77, 2022. https://doi.org/10.56578/jimd010108.

Abstract: For the integrated optimization of job-shop production scheduling and predictive maintenance, this paper fully considers such constraints as product delivery time and changing machine failure rate, and establishes a multi-objective optimization model aiming to minimize the processing cost and the product processing time. The model includes the changing machine failure rate into the integrated optimization of job-shop production scheduling and predictive maintenance, and enables the prediction of the machine state according to the processing time of the current job, laying the basis for the decision-making of the machine activity and the reasonable and effective production planning. In addition, the non-dominated sorting genetic algorithm (NSGA)-II was designed to solve the proposed model. The algorithm performance was improved through the operator crossover and mutation by the simulated binary crossover algorithm (SBX). The proposed strategy was verified through a case study.

Keywords: Production scheduling; Predictive maintenance; Multi-objective optimization; Lead time; Non-dominated sorting genetic algorithm (NSGA)-II

1. Introduction

Production scheduling and equipment maintenance are two important activities in manufacturing job-shops. In actual production, the production schedule often cannot reach the expected effect. One of the main reasons is that the performance degradation and fault maintenance of the machines are not considered when the plan is formulated [1, 2]. The combination of predictive maintenance and production scheduling can ensure the normal operation of the production schedule, improve the job-shop processing efficiency, enhance product quality, effectively control the machine maintenance cost, and realize the integration and reasonable allocation of internal resources of the enterprise [3, 4]. It is of great significance to study the production scheduling and maintenance integration optimization of machine predictive maintenance. The relevant research helps to improve the competitiveness of enterprises, promote the development of manufacturing, and accelerate the intelligent transformation of enterprises [5, 6]. Considering equipment maintenance activities, it is now a research hotspot of production scheduling to integrate production scheduling with equipment maintenance scheduling.

Liao et al. [7] proposed a maintenance strategy based on the number of operations on each machine. A machine is maintained when the number of operations reaches the upper limit. Mosheiov and Sidney [8] proposed the maintenance periodicity, that is, the maintenance activities of the machine should be combined with the working time and machine state. The longer it is until the failure occurs, the more expensive the time spent on maintenance. With minimum processing time as the goal, Cassady et al. [9] established an integrated optimization model of production scheduling and maintenance on a single machine, and compared the differences between integrated scheduling and non-integrated scheduling. Considering the strategies of perfect maintenance and imperfect maintenance, Chung et al. [10] applied the genetic algorithm to solve the job-shop scheduling problem based on the two strategies. Pan et al. [11] proposed a single-machine model based on production scheduling and predictive

maintenance, which can monitor and evaluate the condition of the machine at any time, and introduced the effective life and remaining maintenance life of the machine to describe the degradation degree of the machine.

Targeting the parallel machine, Berrichi et al. [12] tried to minimize the maximum availability through optimization, and applied the multi-objective ant colony optimization (ACO) to solve the ensemble optimization problem. Lu et al. [13] applied minimized maintenance and preventive maintenance for a single machine with random failure of Weibull distribution. Chung et al. [14] took account of perfect maintenance and imperfect maintenance, and applied genetic algorithm to solve the job-shop scheduling problem based on perfect and imperfect maintenance. Ben et al. [15] integrated production and maintenance activities in the job-shop, and proposed an elite genetic algorithm for multi-objective problems, which optimizes and solves the model with completion time as the objective function. In the scheduling research of integrated preventive maintenance, Wang and Liu [16] applied multi-objective genetic algorithm to solve the single-machine problem with maintenance cost, maximum completion time, weighted completion time, delay penalty and equipment utilization as the optimization objectives at the same time. Nima and Hossein [17] implemented the genetic algorithm to solve the hierarchical maintenance scheme model in parallel machine operations.

In actual processing and production, product orders have corresponding delivery dates, which not only limit the processing time of jobs, but may also generate additional processing costs [18]. Therefore, product delivery time is a factor that cannot be ignored in the research on integrated job-shop maintenance and scheduling. This paper introduces the factors of product delivery time and cost, and incorporates the machine failure rate into the job-shop production scheduling with predictive maintenance. Considering the changing failure rate of the machine, a multi-objective integrated optimization model of the job-shop was established in the light of predictive maintenance, and a genetic solution algorithm was designed to solve the model. On this basis, the reasonable production plan was chosen according to different decision-making needs.

The remainder of this paper is organized as follows: Section 2 sets up establishes a multi-objective integrated scheduling model considering delivery time; Section 3 designs the non-dominated sorting genetic algorithm (NSGA)-II was designed to solve the multi-objective optimization model; Section 4 uses examples to analyze the model, and verify the designed algorithm; Section 5 summarizes the research conclusions.

2. Model Construction

The job-shop contains m machines, each executes different operations. The jobs must be processed by every machine, yet following different operation sequence. Then, the following sets are defined: the set of jobs to be processed J_k, and the set of machines M_m. At the beginning of machine processing, all parts have arrived at the job-shop. Due to the limitation of the delivery period, the predictive maintenance decisions of machines must consider the maintenance cost and time of the entire machine, and the delivery time of the jobs to the processed, as well as the penalty cost of early and late completion of the jobs.

Figure 1 shows the delivery time and fault constraint. According to the model prediction, the machine belongs to the dangerous interval at time A, and needs perfect maintenance. However, the perfect maintenance at this moment will cause the complete date of job J_2 to exceed the specified delivery period, creating an additional processing cost. If imperfect maintenance is carried out here, the processing time and maintenance cost will rise, and job J_2 can be completed before the delivery period, thereby avoiding the penalty cost of late completion.

This paper proposes a multi-objective integrated optimization decision-making model. While carrying out the maintenance activities of the machine, the constraints of the machine state and the job delivery time were considered to minimize the processing time and optimize the processing cost. Multiple variables are introduced (Table 1) to build up the model.

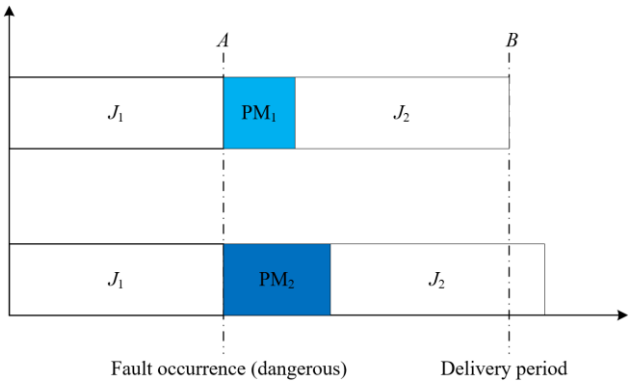

Figure 1. Delivery time and fault constraint

Table 1. Model parameters

Variables	Meaning
J_i	i=1,2…k, the set of jobs to be processed, k is the total number of jobs
O_{ip}	p=1,2…n, the processing sequence of job J_i
M_m	m=1,2…q, the set of machines, q is the total number of machines
P_m	m=1,2…q, the time of perfect maintenance of machine M_m
P'_m	m=1,2…q, the time of imperfect maintenance of machine M_m
U_m	Decision variable, the total number of processing operations of machine M_m
t_{ipm}	Processing time of operation p of job J_i on machine m.
X_{ipm}	Decision variable, if operation p of job J_i on machine M_m, then X_{ipm} =1; otherwise, X_{ipm} =0.
S^w_{ims}	The start time of job J_i being processed at the s-th station of machine M_m.
E^w_{ims}	The end time of job J_i being processed at the s-th station of machine M_m.
W_{ms}	The waiting time of machine m after the s-th processing activity.
T_{ms}	The processing time of the s-th processing activity of machine M_m.
N_{ms}	The number of perfect maintenances of machine M_m for the s-th processing activity.
N'_{ms}	The number of imperfect maintenances of machine M_m for the s-th processing activity.
PM_1	Primary maintenance strategy (imperfect maintenance)
PM_2	Secondary maintenance strategy (perfect maintenance)
λ	Fault attenuation coefficient of perfect maintenance
R	Predicted machine state
D_m	Fault rate of machine M_m
$D_{[e]}$	Threshold of fault rate
L_i	Product delivery period of job J_i
C_{PM}	Maintenance cost coefficient of stage n maintenance strategy
α	Cost operation coefficient of machine processing
α_1	Cost operation coefficient of machine waiting time
β	Early delivery penalty coefficient of machine
β_1	Late delivery penalty coefficient of machine

2.1 Objective function

The costs being considered mainly include the processing cost of machine, and the early/late delivery penalty costs.

2.1.1 Processing cost

The processing cost includes the processing operation cost of the machine and the maintenance cost of the machine. Whenever a machine performs a processing task, it will cause a certain cost loss; when the machine is turned on and idle, it will also incur costs; different maintenance methods generate different maintenance costs. The processing cost can be modeled as:

$$C_1 = \sum_{m=1}^{q}\left\{\left[\sum_{i=1}^{k}\sum_{p=1}^{n}(t_{ipm}\cdot X_{ipm}\cdot\alpha)\right]+\sum_{s=1}^{U_m-1}W_{ms}\cdot\alpha_1+\sum_{s=1}^{U_m-1}(G_{ms}\cdot P_m\cdot c_{PM_1}+G'_{ms}\cdot P'_m\cdot c_{PM2})\right\} \tag{1}$$

2.1.2 Penalty costs

Based on the just-in-time system, when the finished job exceeds the given delivery time or is delivered before the delivery time, certain resources will be consumed, and this part of the consumed resources is called penalty cost. Given the early time delivery penalty coefficient β and the late delivery penalty coefficient β_1, the penalty cost model can be established as:

$$\begin{aligned} C_2 = \min\sum_{i}^{k}\Big\{&\beta\cdot\max\left(L_i-\max\left\{E^w_{ipm}\mid p\in[1,n],m\in[1,q]\right\},0\right)+\\ &\beta_1\cdot\max\left(\max\left\{E^w_{ipm}\mid p\in[1,n],m\in[1,q]\right\}-L_i,0\right)\Big\}\end{aligned} \tag{2}$$

Based on the above analysis and definition, the minimum cost and minimum processing time can be respectively expressed as:

$$\begin{aligned} C = &\sum_{m=1}^{q}\left\{\left[\sum_{i=1}^{k}\sum_{p=1}^{n}(t_{ipm}\cdot X_{ipm}\cdot\alpha)\right]+\sum_{s=1}^{U_m-1}W_{ms}\cdot\alpha_1+\sum_{s=1}^{U_m-1}(G_{ms}\cdot P_m\cdot c_{PM_1}+G'_{ms}\cdot P'_m\cdot c_{PM2})\right\}\\ &+\min\sum_{i}^{k}\left\{\beta\cdot\max\left(L_i-\max\left\{E^w_{ipm}\mid p\in[1,n],m\in[1,q]\right\},0\right)+\beta_1\cdot\max\left(\max\left\{E^w_{ipm}\mid p\in[1,n],m\in[1,q]\right\}-L_i,0\right)\right\}\end{aligned} \tag{3}$$

The minimum processing time can be expressed as:

$$T = \max\left\{\sum_{i=1}^{k}\sum_{p=1}^{n}\left(X_{ipm}\cdot t_{ipm}\right)+\sum_{s=1}^{U_m-1}\left[W_{ms}+(G_{ms}\cdot P_m+G'_{ms}\cdot P'_m)\right]\middle| m=1,2,...q\right\} \tag{4}$$

2.2 Processing constraints

The relevant constraints are as follows:
A job can only be processed on one machine at a time:

$$\sum_{i=1}^{k}X_{ipm}=1\ \forall p,m \tag{5}$$

Each machine can only process one job at a time:

$$\sum_{p=1}^{n}X_{ipm}=1\ \forall i,m \tag{6}$$

Each machine can only execute on maintenance activity at a time:

$$Gms+G'_{ms}\le 1 \tag{7}$$

The start time of machine processing cannot precede the end time of machine maintenance:

$$E^{w}_{ims}+\max\{P_m\cdot G_{ms'}\cdot P_m\cdot G_{ms}\}\le S^{w}_{i'ms+1} \tag{8}$$

Machine processing cannot be interrupted:

$$E^{w}_{ims}=S^{w}_{ims}+T_{ms}\ \ \forall i,m,s \tag{9}$$

The start time of machine processing cannot precede the end time of the previous operation:

$$E^{w}_{ims-1}\le S^{w}_{i'ms}\ \ \forall i,m,s \tag{10}$$

The start time of machine processing cannot precede the end time of processing of the job on the previous machine:

$$E^{w}_{ims-1s}\le S^{w}_{ims}\ \ \forall i,m,s \tag{11}$$

3. Algorithm Design

For the established multi-objective optimization model, the multi-objective genetic algorithm was elected to solve the model. The multi-objective genetic algorithm is updated and optimized on the genetic algorithm. Many effective algorithms have been formulated by combining the concept of Pareto front solution set with the multi-objective genetic algorithm [19, 20]. Among them, the NSGA is the most representative one.

Step 1. Create initial variables, and set the number of iterations *Items*=1, the maximum number of iterations *M*, the population size *N*, the mutation probability Q_m and the crossover probability P_m.

Step 2. Initialize the population by randomly generating samples.

Step 3. Computing the individual fitness S_m in the population.

Step 4. If *Items* > 1, replace the individual with the lowest fitness of the new population with the individual with the highest fitness in the previous generation to ensure the population quality; otherwise, go to Step 5.

Step 5. If *Items* < *M*, retain the individual with the highest fitness in the population; otherwise, terminate the program, and output the best individual from the previous population as the optimal solution.

Step 6. Randomly generate the crossover probability P. If $P<P_m$, randomly crossover two individuals other than the best individual; otherwise, go to Step 7.

Step 7. Randomly generate the mutation probability Q. If $Q<Q_m$, randomly mutate individuals other than the best individual; otherwise, go to Step 8.

Step 8. Generate a new population by merging the parent and child populations.

Step 9. Carry out fast non-dominated sorting, and calculate crowding degree and fitness.

Step 10. Select the best individuals via tournament method and form a new population. Let *Items*=*Items*+1. If *Items*<*M*, go back to Step 3; otherwise, output the final results.

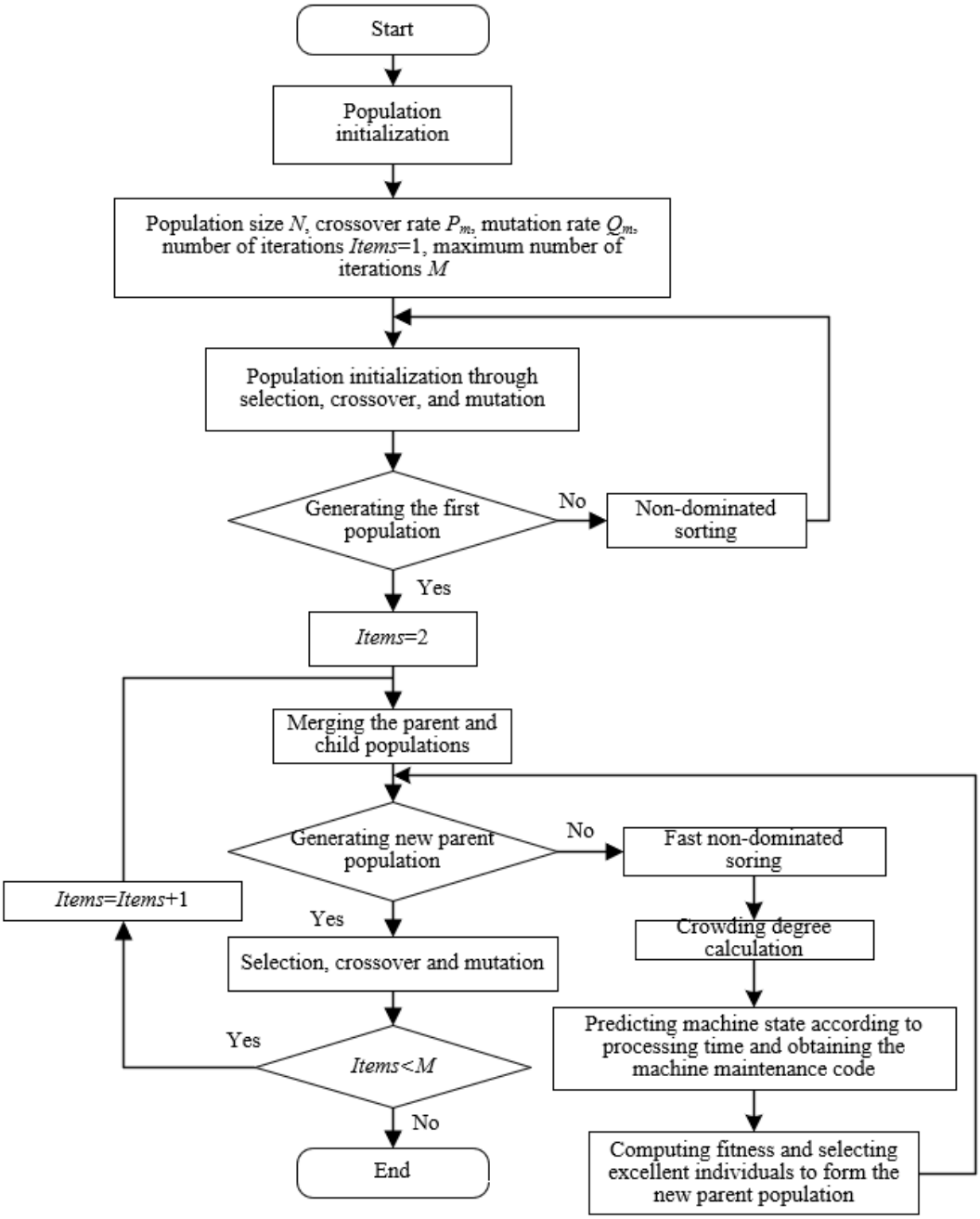

Figure 2. Flow chart of NSGA-II

The coding, crossover and mutation links of NSGA-II can be designed as follows (Figure 2):

3.1 Coding

In the process of solving the genetic algorithm, the coding and decoding rules are the basics. Algorithm coding effectively transforms actual parameters into algorithm units, which can better complete the solution of the objective function. A reasonable coding method can greatly improve the convergence speed and solution efficiency of the algorithm.

In the initial stage of encoding, double-layer encoding is adopted. The first layer represents the number sequence of the jobs processed by the machine. For example, {6,1,2,5,3,4} means that the jobs processed by the machine are: job 6→job 1→job 2→job 5→job 3→job 4. The second layer indicates whether the machine is maintained after the job is processed. 0 means no maintenance, 1 means minor repairs, and 2 means normal maintenance. For example, {0,1,0,0,2,0} means that minor repairs are performed after the first job is processed, and normal maintenance operations are performed after the second job is processed (Figure 3).

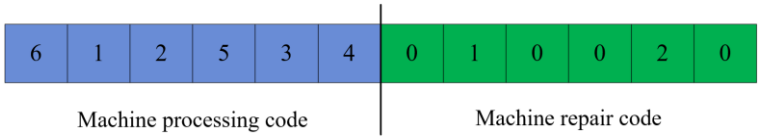

Figure 3. Coding rules

3.2 Crossover

The SBX crossover algorithm is adopted to solve the multi-objective problem. This crossover algorithm simulates single-point binary, and is often used to solve the multi-objective model of real number coding. Let $P_1=\{P_1^1,P_2^1.....P_N^1\}$ and $P_2=\{P_1^2,P_2^2.....P_N^2\}$ be two parent individuals. After crossover of them, two new individuals C_1 and C_2 can be generated by:

$$C_i^1=0.5\cdot\left[P_i^1+P_i^2+\beta\left(P_i^1-P_i^2\right)\right] \tag{12}$$

$$C_i^2=0.5\cdot\left[P_i^1+P_i^2+\beta\left(P_i^2-P_i^1\right)\right] \tag{13}$$

where, β is the diffusion factor, which can be dynamically generated by:

$$\beta=\begin{cases}\left(2\cdot random\right)^{\frac{1}{1+\eta}} & random\le 0.5\\ \left(\dfrac{1}{\left(1-2\cdot random\right)}\right)^{\frac{1}{1+\eta}} & random>0.5\end{cases} \tag{14}$$

η is a parameter defined by the algorithm. The greater the η value, the greater the similarity between the generated child and the parent after the crossover. The search performance of the entire algorithm is controlled by adjusting the value of η. During the crossover, SBX has the following characteristics:

After binary decoding, the decoding average of the parent and child is the same: $C_1+C_2=P_1+P_2$ (Figure 4):

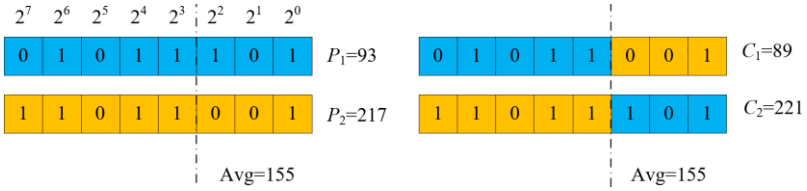

Figure 4. SBX crossover

The quotient of the difference between the decoded values after the crossover is slightly close to 1.

The difference quotient β measures the difference of the decoded values before and after the crossover:

$$\beta=\left|\frac{C_1-C_2}{P_1-P_2}\right| \tag{15}$$

According to the above two characteristics, the child can be inversely solved according to the value of the parents:

$$C_1=0.5\cdot\left(P_1+P_2\right)-0.5\cdot\beta\cdot\left(P_2-P_1\right) \tag{16}$$

$$C_2 = 0.5 \cdot (P_1 + P_2) + 0.5 \cdot \beta \cdot (P_2 - P_1) \quad (17)$$

Different child generations can be generated with different β values, which ensure the flexibility and diversity of the population.

3.3 Mutation

In the multi-objective optimization algorithm, polynomial mutation is widely used as a common mutation method. Polynomial mutation is also applicable to real-coded multi-objective algorithm solving. The encoding is divided into two parts: the first part adopts the real number encoding method, and the second part adopts the binary encoding method. For the mutation encoded in the first part, the authors chose the polynomial random mutation method. The binary encoding of the second part is the normal variation.

Let $X_k^1 = \{1,2 \ldots K\}$ be the first layer of codes. A random number n between [1, K] is randomly generated to determine the mutation position. Then, a random integer m is generated in [1, K/2], and $X_n^1 + m$ is taken as the mutated value. If $X_n^1 + m$ is beyond the coding range, $X_n^1 - m$ will be treated as the mutated value. Next, search for the position index i of the mutated value, and make $X_i^{1=} X_n^1$ to complete the mutation operation.

The second layer adopts binary encoding. After improvement, the second layer of coding does not have unique coding, and can directly carry out code mutation.

The codes are divided into three levels, 0 for normal processing, 1 for minor repairs and 2 for normal maintenance. Three groups of mutation probabilities are defined, P1, P2, and P3. Each time a mutation operation is performed, each code in the second layer has a certain probability to mutate and generate a new code information node (Figure 5).

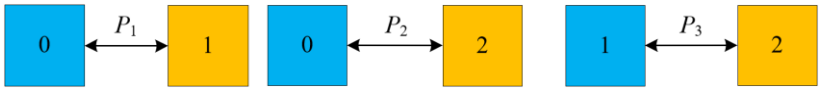

Figure 5. Mutation operation

When performing the mutation operation, each code element of the second layer coding has a certain mutation probability, i.e., the code element 0 has a P1 probability to mutate into 1, and similarly, the code element 1 also has a P1 probability to mutate into 0.

3.4 Selection

According to the fitness function of the algorithm, individuals are randomly selected from the parent population and retained to the next generation, while the remaining individuals are eliminated. Common selection strategies include roulette, random sampling, and tournament method. The tournament method selects a certain number of relatively superior individuals from the population to enter the tournament, which is suitable for solving the max-min optimization problems. The specific steps are as follows:

Step 1. Select the number of individuals according to the percentage of the population each time.

Step 2. Randomly select individuals from the population for fitness sorting, and retain individuals with higher fitness the next generation.

Step 3. Repeat Step 2 until the obtained individuals can form a new generation population.

4. Case Analysis

The case analysis focuses on a job-shop which intends to process six jobs, each of which needs to go through six operations. The operation sequence varies from job to job. The processing time is shown in Table 2. The machine failure rates obey the Weibull distribution, β=3.738 and η=927.535.

Table 2. Processing schedule

Job number	Machine number, processing time (min)					
J_1	(1,9)	(2,40)	(3,75)	(4,79)	(5,11)	(6,71)
J_2	(1,40)	(2,48)	(3,97)	(4,84)	(5,123)	(6,42)
J_3	(1,41)	(2,43)	(3,92)	(4,89)	(5,13)	(6,95)
J_4	(1,58)	(2,52)	(3,51)	(4,33)	(5,91)	(6,108)
J_5	(1,47)	(2,29)	(3,53)	(4,51)	(5,42)	(6,13)
J_6	(1,78)	(2,53)	(3,90)	(4,108)	(5,39)	(6,14)

Table 3. Schedule of different maintenance activities

Maintenance level	PM1	PM2
Time, min	30	40

Table 4. Delivery time constraints

Job	J_1	J_2	J_3	J_4	J_5	J_6
Delivery time, min	456	537	498	502	560	423

Table 5. Early/late delivery penalties

Job	J_1	J_2	J_3	J_4	J_5	J_6
Early penalty coefficient, %	20	20	20	20	20	20
Late delivery penalty coefficient, %	30	30	30	30	30	30

Table 6. Machine cost coefficient

Machine	M_1	M_2	M_3	M_4	M_5	M_6
Processing cost coefficient, %	20	25	15	20	25	20
Waiting time coefficient, %	10	15	10	15	15	15

The maintenance level is divided according to the processing state of the machine, and the maintenance of the machine is divided into two levels PM1 and PM2. Different maintenance activities are carried out according to the operation of the machine. The maintenance schedule is shown in Table 3. The processing costs and early/late delivery penalty coefficients are shown in Tables 4 and 5, respectively. The maintenance cost coefficients caused by the maintenance activities is uniformly set to 20% (Table 6).

The mathematical model is established and solved by NSGA-II algorithm, which is written by MATLAB. The program is run on MATLAB R2014a. The relevant parameters are configured as follows: the population size N=300, the number of iterations 500, the mutation probability Q_m within [0.95, 1), and the crossover probability P_m within [0.001, 0.05). After repeating the program 10 times, the Pareto front is shown in Figure 6.

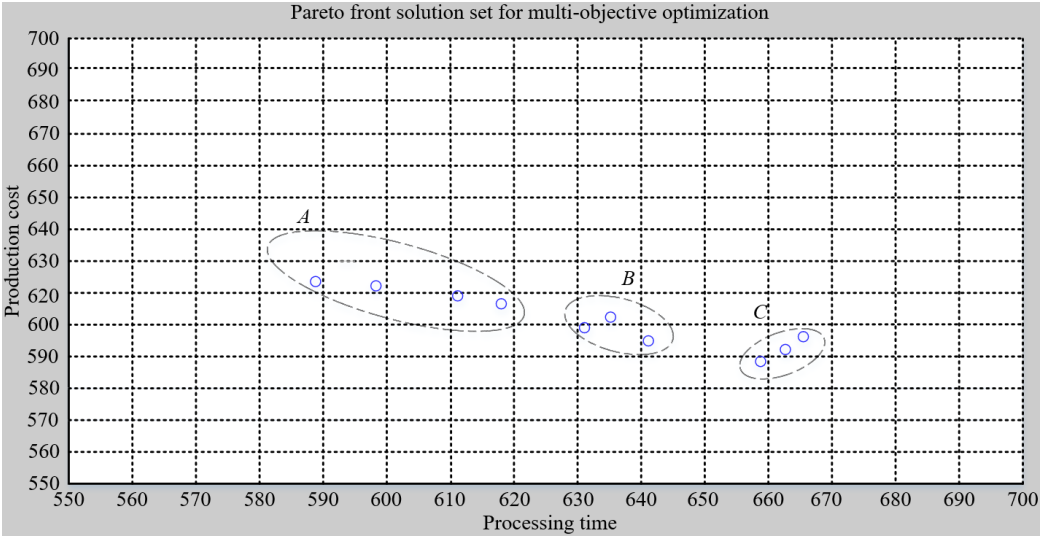

Figure 6. Pareto Front

Table 7. Calculation results

Group	T1	T2	T3	T4	T5
Processing time, min	589	598	611	618	631
Processing cost	623.1	622.1	619.1	617.25	608.7
Group	**T6**	**T7**	**T8**	**T9**	**T10**
Processing time, min	635	641	659	663	665
Processing cost	611.9	604.8	598.8	602.3	605.8

The calculation data are sorted out as in Table 7. For multi-objective optimization problems, the objective optimization functions restrict each other, making it impossible to derive the absolute optimal solution. The effective solutions can only be obtained according to different objectives in the feasible solution set. This is to allocate the job-shop resources reasonably, so that the processing time and processing cost of the tasks can reach a relative balance. The optimal Gantt charts of job processing time and production cost are drawn respectively in the ten test results, as shown in Figures 7 and 8.

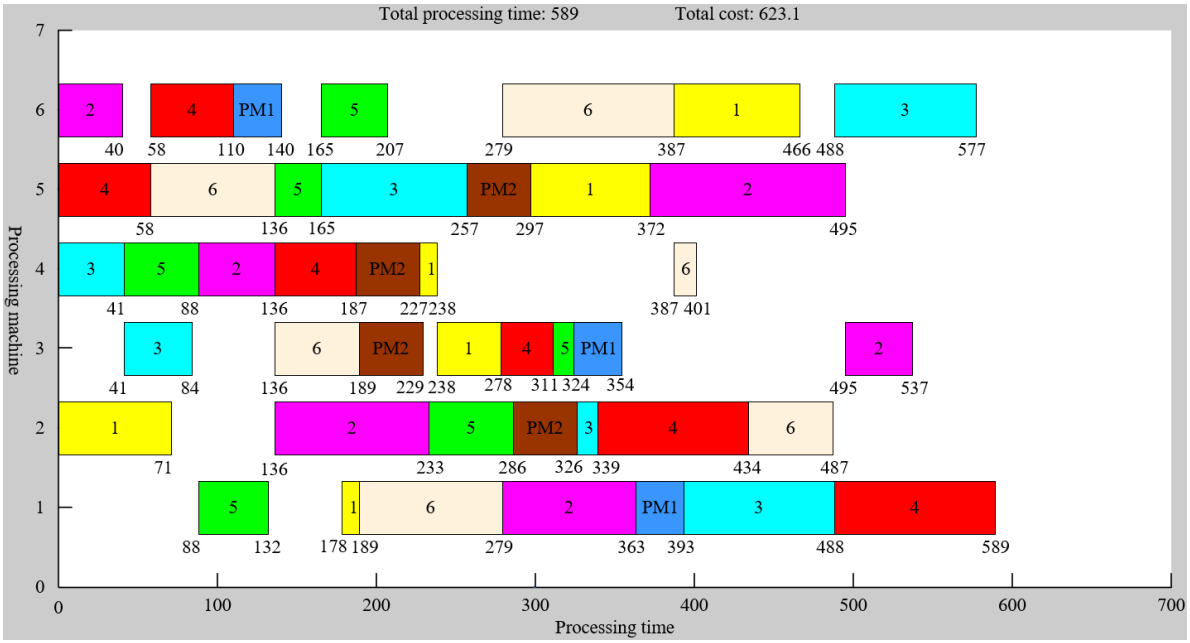

Figure 7. Optimal Gantt chart of job processing time

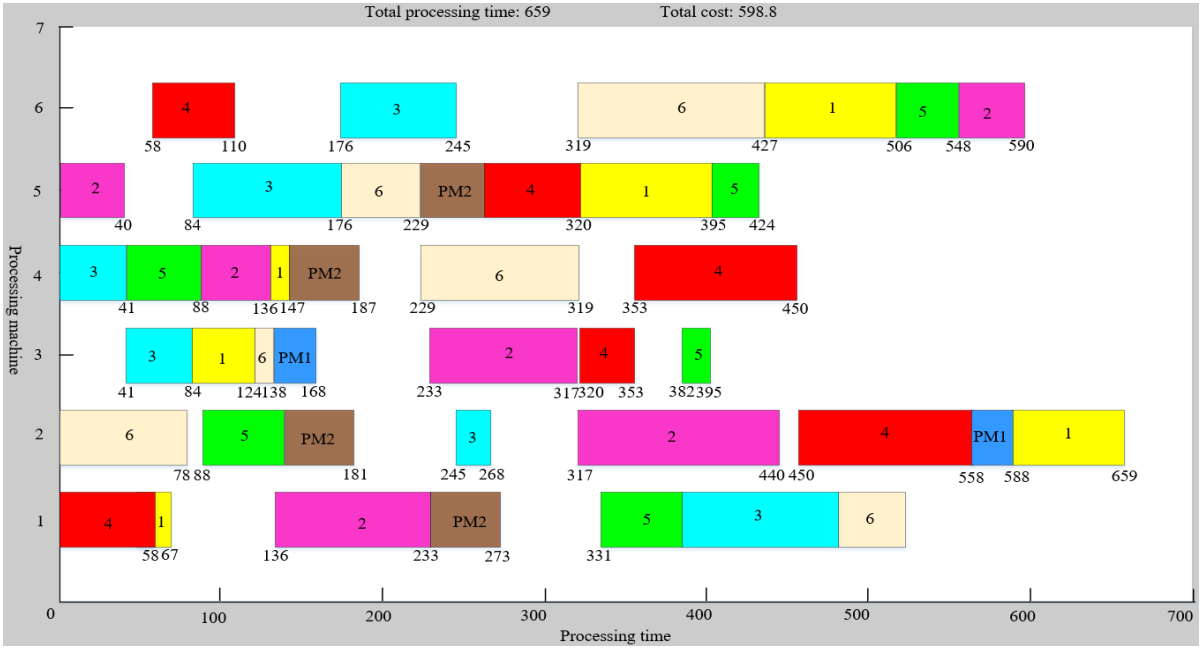

Figure 8. Optimal Gantt chart of processing cost

It can be seen from the experimental data that the ideal and optimal result does not exist under the interaction between the processing time and the total processing cost. When the machining time is short, the processing cost of the machine is low, but it may result in a high early/late delivery penalty, which increases the cost. When the processing time is long, the penalty costs should be minimized, but the idle time of the machine will be too long, resulting in a waste of resources. Each solution constrains each other between effective values and achieves a relative balance. Therefore, the solutions in the solution set are effective and have a certain reference value.

In Figure 6, a multi-objective Pareto solution set has been generated. According to Pareto theory, all solutions in the solution set are considered to be valid and feasible solutions. Different solutions are selected as the optimal target solutions according to different decision-making requirements. Decision-making is divided into the following three categories.

(1) The job-shop scheduling decision is biased towards shortening the processing time, and the solutions in the Pareto solution set an area can be used as the target optimal solution.

(2) The job-shop scheduling decision is biased towards reducing the production cost, and the solutions in the Pareto solution set C area can be used as the target optimal solution.

(3) The job-shop scheduling decision tends to balance processing time and production costs, and the solutions in the Pareto solution set B area can be used as the target optimal solution.

5. Conclusions

Taking the job-shop as the research object, this paper clarifies the importance of the delivery time condition in actual job-shop processing, and discusses the integration of production scheduling and machine maintenance under the constraints of the delivery date. Specifically, the authors established a multi-objective optimization model with the goal of minimizing the total processing cost, and optimized and solved the model by NSGA-II. In addition, the operations of the algorithm were designed in details: The algorithm performance was enhanced by SBX-guided crossover and mutation. The model was verified through a set of examples. The multi-objective Pareto solution set was obtained, and the accuracy of the Pareto solution set was analyzed, which further proves the feasibility and effectiveness of the model. The proposed method can effectively reduce time delay and resource waste in job-shop production scheduling, and reduce equipment maintenance cost and maintenance workload.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflict of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. Abedi, R. Chiong, N. Noman, and R. Zhang, "A multi-population, multi-objective memetic algorithm for energy-efficient job-shop scheduling with deteriorating machines," *Expert. Syst. Appl.*, vol. 157, Article ID: 113348, 2020. https://doi.org/10.1016/j.eswa.2020.113348.

[2] S. Chansombat, P. Pongcharoen, and C. Hicks, "A mixed integer linear programming model for integrated production and preventive maintenance scheduling in the capital goods industry," *Int. J. Prod. Res.*, vol. 57, no. 1, pp. 61-82, 2018. https://doi.org/10.1080/00207543.2018.1459923.

[3] Y. Li, M. S. Yang, X. Chen, Z. P. Zhan, E. B. Xu, and Y. Li, "Integrated optimization of predictive maintenance and production scheduling for flow-shop," *Mech. Sci. Technol. Aerospace Eng.*, vol. 41, no. 7, pp. 1055-1061, 2022.

[4] X. G. Zhang, Y. Q. Yin, and C. C. Wu, "Scheduling with non-decreasing deterioration jobs and variable maintenance activities on a single machine," *Eng. Optimiz.*, vol. 49, no. 1, pp. 84-97, 2017. https://doi.org/10.1080/0305215X.2016.1163629.

[5] E. Moradi, S. M. T. Fatemi Ghomi, and M. Zandieh, "Bi-objective optimization research on integrated fixed time interval preventive maintenance and production for scheduling flexible job-shop problem," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7169-7178, 2011. https://doi.org/10.1016/j.eswa.2010.12.043.

[6] A. Khatab, E. L. H. Aghezzaf, C. Diallo, and I. Djelloul, "Selective maintenance optimisation for series-parallel systems alternating missions and scheduled breaks with stochastic durations," *Int. J. Prod. Res.*, vol. 55, no. 9-10, pp. 3008-3024, 2017. https://doi.org/10.1080/00207543.2017.1290295.

[7] C. J. Liao, C. M. Chen, and C. H. Lin, "Minimizing makespan for two parallel machines with job limit on each availability interval," *J. Oper. Res. Soc.*, vol. 58, no. 7, pp. 938-947, 2020. https://doi.org/10.2307/4622787.

[8] G. Mosheiov and J. B. Sidney, "Scheduling a deteriorating maintenance activity on a single machine," *J. Oper. Res. Soc.*, vol. 61, no. 5, pp. 882-887, 2010. https://doi.org/10.1057/jors.2009.5.

[9] C. R. Cassady and E. Kutanoglu, "Integrating preventive maintenance planning and production scheduling for a single machine," *IEEE T. Reliab.*, vol. 54, no. 2, pp. 304-309, 2005. https://doi.org/10.1109/TR.2005.845967.

[10] S. H. Chung, F. T. S. Chan, and H. K. Chan, "A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling," *Eng. Appl. Artif. Intel.*, vol. 22, no. 7, pp. 1005-1014, 2009. https://doi.org/10.1016/j.engappai.2008.11.004.
[11] E. Pan, W. Liao, and L. Xi, "Single-machine-based production scheduling model integrated preventive maintenance planning," *Int. J. Adv. Manuf. Tech.*, vol. 50, no. 1-4, pp. 365-375, 2010. https://doi.org/10.1007/s00170-009-2514-9.
[12] A. Berrichi, F. Yalaoui, L. Amodeo, and M. Mezghiche, "Bi-objective ant colony optimization approach to optimize production and maintenance scheduling," *Comput. Oper. Res.*, vol. 37, no. 9, pp. 1584-1596, 2010. https://doi.org/10.1016/j.cor.2009.11.017.
[13] Z. Q. Lu, W. W. Cui, and X. L. Han, "Integrated production and preventive maintenance scheduling for a single machine with failure uncertainty," *Comput. Ind. Eng.*, vol. 80, no. 2, pp. 236-244, 2015. https://doi.org/10.1016/j.cie.2014.12.017.
[14] S. H. Chung, F. T. S. Chan, and H. K. Chan, "A modified genetic algorithm approach for scheduling of perfect maintenance in distributed production scheduling," *Eng. Appl. Artif. Intel.*, vol. 22, no. 7, pp. 1005-1014, 2009. https://doi.org/10.1016/j.engappai.2008.11.004.
[15] M. Ben Ali, M. Sassi, M. Gossa, and Y. Harrath, "Simultaneous scheduling of production and maintenance tasks in the job shop," *Int. J. Prod. Res.*, vol. 49, no. 13, pp. 3891-3918, 2011. https://doi.org/10.1080/00207543.2010.492405.
[16] S. J. Wang and M. Liu, "Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning," *J. Manuf. Syst.*, vol. 37, pp. 182-192, 2015. https://doi.org/10.1016/j.jmsy.2015.07.002.
[17] M. S. Nima and I. Hossein, "A scheduling model for serial jobs on parallel machines with different preventive maintenance," *Int. J. Adv. Manuf. Technol.*, vol. 70, no. 9-12, pp. 1579-1589, 2014. https://doi.org/10.1007/s00170-013-5348-4.
[18] M. Assid, A. Gharbi, and A. Hajji, "Joint production, setup and preventive maintenance policies of unreliable two-product manufacturing systems," *Int. J. Prod. Res.*, vol. 53, no. 15, pp. 4668-4683, 2015. https://doi.org/10.1080/00207543.2015.1030468.
[19] S. Sivasubramani and K. S. Swarup, "Multi-objective harmony search algorithm for optimal power flow problem," *Int. J. Elec. Power.*, vol. 33, no. 3, pp. 745-752, 2011. https://doi.org/ 10.1016/j.ijepes.2010.12.031.
[20] S. Verma, M. Pant, and V. Snasel, "A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems," *IEEE Access*, vol. 9, pp. 57757-57791, 2021. https://doi.org/10.1109/ACCESS.2021.3070634.