Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 10, No. 3, 2023

Event-driven dynamic Job-shop scheduling method with strong process constraints

Chenlu Zhang, Ruijuan Zheng

School of Henan University of Science and Technology, Luoyang 471000, China

Abstract: In the actual production environment, there are many uncertainties that need to be handled. Since the occurrence time and frequency of dynamic events cannot be predicted, frequent global scheduling calculation will cause waste of time and resources. Therefore, facing the challenge of job shop scheduling problem in dynamic scenarios, this paper studied the dynamic event-driven job shop scheduling problem with complex process constraints, and proposed an improved NSGAII algorithm. At the end of this paper, the designed dynamic scheduling method is applied to a variety of benchmarks through experiments to verify the effectiveness of the proposed method in solving the job shop scheduling problem in a dynamic environment.

Keywords: Flexible Job-shop; NSGA-II; Dynamic event-driven.

1. Introduction

With the continuous improvement of the information level of manufacturing enterprises, the traditional production workshop is gradually moving towards automation and intelligence [1]. The job shop scheduling problem belongs to the Flexible Job Shop Scheduling Problem (FJSP), which has been widely concerned by academia and industry in recent years. The FJSP problem consists of two sub-problems: machine allocation and process sequencing. The former is to select a production equipment for each process from the equipment candidate set, and the latter is to schedule all processes on all devices sequentially to obtain a satisfactory production plan. This is one of the most critical issues in the production planning and management of manufacturing processes in manufacturing enterprises. The FJSP problem is very complex and has been proved to be an NP-Hard problem [2]. Due to the complexity of such problems, it is difficult to solve them in a reasonable time using traditional mathematical optimization methods. Therefore, for manufacturing enterprises, it is of great practical significance to establish scientific production planning and in-depth study scheduling model and solve it effectively [3].

The manufacturing system is often accompanied by some unpredictable unstable factors, such as personnel transfer, machine failure and so on. Therefore, to ensure the smooth progress of workshop production, it is necessary for production scheduling plan to make timely and effective local dynamic adjustment of unstable factors. Zhang et al. [4] considered the dynamic environment and carried out the automatic evolution of heuristic scheduling based on genetic programming through feature extraction to achieve scheduling within the local minimum range. An et al. [5] established a machine preventive maintenance model to determine the machine maintenance plan and proposed an improved adaptive reference vector non-dominated sorting genetic algorithm III to solve the problem from the perspective of preventive scheduling. Zhang et al. [6] built a deep reinforcement learning to construct a multi-agent scheduling system for large-scale order and resource random disturbance scenarios, which can effectively deal with a variety of unpredictable abnormal events in the workshop but cannot solve the scheduling problem with task priority. Wen et al. [7] proposed an improved genetic algorithm to solve the problem of dynamic integrated process planning and task scheduling and achieved better scheduling results. However, this algorithm can only deal with dynamic events of single equipment failure type.

Above research status, the research content is mostly a single target and an idealized production scene. There are few studies on production scheduling problems with strong process constraints, especially the dynamic job-shop scheduling problem. Therefore, it is expected to achieve great results to study how to construct an efficient dynamic production task scheduling model, solve various difficult elements and emergencies encountered in the process of workshop production and operation, and promote industrial production to intelligent construction such as flexible manufacturing and intelligent manufacturing [8]. Therefore, this paper conducts research based on the above content to provide theoretical and technical support for workshop production task scheduling, promote the utilization and integration of various resources in industrial manufacturing scenarios, and provide a solid foundation for the innovation and development of intelligent manufacturing enterprises.

2. Problem formulation

2.1. Problem model

Considering the response method of scheduling problem with complex process under the influence of dynamic events, it is to study how to find a stable and effective scheduling method based on the existing initial scheduling scheme under the influence of production state changes. In the workshop, there are three main types of dynamic events that occur most frequently, including urgent order arrival, human priority promotion of unfinished work, and production delay caused by machine failure. When a dynamic event occurs in the shop floor, it is necessary to apply the corresponding rescheduling strategy according to the type of event to generate different rescheduling schemes.

Urgent order arrival: During the process of processing in the workshop, new orders will still arrive continuously. Some newly arrived orders have an earlier delivery date than some original orders, so they are relatively more urgent. In the rescheduling stage, the production of these orders needs to be planned. Currently, the rescheduling faces the problem of inserting orders in the planned machining tasks.

Priority adjustment of unfinished tasks: When the workshop is producing according to the planned scheduling scheme, some tasks face the situation that they need to deliver workpieces in advance. If the production is carried out according to the current established plan, it cannot meet the temporary change of enterprise production demand. Therefore, it is necessary to improve the production priority of this part of the workpiece, to meet the delivery date requirements of the enterprise.

Machine failure: Because the production of special robots requires the use of high-precision equipment to process the workpiece, it is easy to appear in the process of equipment wear, accuracy decline and other events leading to equipment failure. When the machine breaks down, the workshop staff is required to troubleshoot and repair the fault in time. During this maintenance period, the equipment will not be able to produce according to the expected scheduling scheme. Therefore, when the machine failure occurs, it is necessary to reschedule in time to reduce the loss caused by the butterfly effect of the machine failure.

The purpose of solving the above problem is to determine the processing path of each job and the start and end time of each operation on the machine tool under the interference of dynamic events, to obtain an efficient and stable rescheduling scheme. To facilitate the subsequent description, the following assumptions are given.

(1) It is assumed that all machines are available at the initial scheduling time.

(2) The time of transportation and equipment loading, unloading or cleaning is not considered.

(3) Machine failures occur randomly.

(4) If the machine fails, it is regarded as repairable within a certain period.

(5) The machine shall be deemed to be ready for use immediately after fault repair.

(6) The unfinished processes caused by machine failure need to be reprocessed.

(7) The mean time between failures and mean time to repair of the same machine follow Poisson distribution.

(8) A machine can process only one job at a time.

(9) Any job can be processed by only one machine at a time.

2.2. Notations

n : Total number of jobs.

e : Total number of machines.

T_0 : Total number of operations.

H : Total number of process routes.

J : $J = J_1, J_2, \ldots, J_n$ is the set of n jobs.

M : $M = M_1, M_2, \ldots, M_e$ is the set of e machines.

J_i : The i -th job in job set.

M_k : The k -th machine in machine set.

$load_k$: Number of operations on machine k .

O_{ij} : The operation j of the job i .

C_i : The completion time of job i .

$ST_{O_{ij}}$: The start time of O_{ij} in initial scheduling.

$st_{O_{ij}}$: The start time of O_{ij} after rescheduling.

$Start_{ij}$: The start time of O_{ij} on M_k .

End_{ij} : The completion time of O_{ij} .

T_k^M : The free time set of M_k .

T_{ij} : The process time of O_{ij} .

E_i : The lead time of job i .

T_i : The delay time of job i .

$$Z_{ij} = \begin{cases} 1, & \text{if the order of } O_{ij} \text{ changes after recheduling} \\ 0, & \text{otherwise} \end{cases}$$

The research goal of this paper is to maximize the production efficiency and production index through reasonable planning of the task working sequence on each equipment for the production tasks with complex process constraints, to optimize the purpose of enterprise production process. The optimization objective is to minimize the maximum completion time, and the calculation formula is shown in Equation (1).

$$f_1 = \min(\max(C_i)) \tag{1}$$

Considering the reduction of order delay default and other consequences caused by rescheduling process, the optimization objective is established to minimize rescheduling time deviation. The calculation formula is shown in Equation (2).

$$f_2 = \min(\sum_{i=1}^{n}\sum_{j=1}^{m}| ST_{ij} - st_{ij} |) \tag{2}$$

In consideration of reducing the additional production burden brought by rescheduling process, the optimization objective is established to minimize the change of task process sequence. The calculation formula is shown in Equation (3).

$$f_3 = \min(\sum_{i=1}^{n}\sum_{j=1}^{m}(1 - Z_{ij})) \tag{3}$$

3. Proposed approach

In this paper, an event-driven dynamic shop scheduling framework is constructed, and a multi-objective hybrid algorithm MNV based on individual density optimization strategy, NSGAII algorithm and variable neighborhood search is designed to solve the rescheduling scheme. Figure 1 shows the overall framework flow of the algorithm.

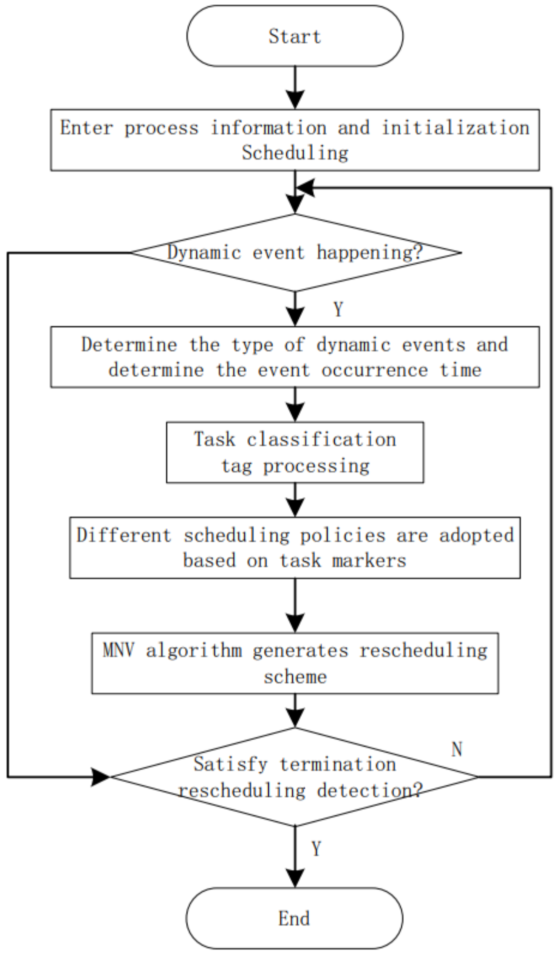


Fig. 1 Event-driven dynamic scheduling framework

3.1. Task classification strategy

The algorithm adopts a process adjustment method based on task classification and adopts different task classification strategies according to different dynamic events. The details are as follows:

Urgent order arrives: When an urgent task arrives, it needs to locate the occurrence time of the event, and the tasks are divided into three categories according to their impact degree.

(1) Direct impact: Tasks with higher priority than urgent tasks.

(2) Indirect effects: Tasks of equal priority.

(3) Others-tasks: The task which priority is higher than the urgent task, and tasks that were in progress when the event occurred.

Priority adjustment of unfinished tasks: When the priority of a task is promoted, the occurrence time of the event needs to be located, and the tasks are divided into two categories according to their impact degree.

(1) Direct impact: Tasks below the adjusted task priority.

(2) Others-tasks: tasks with an adjusted priority of no lower priority, and tasks that were in process when the event occurred.

Mechanical breakdown: When there is a machine failure, it is necessary to locate the time of the failure and the location of the machine and divide the task into three categories according to the degree of influence. The classification process is shown in Figure 2.

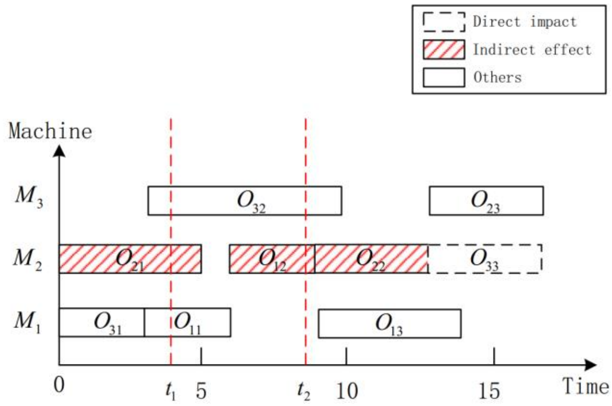


Fig. 2 Production task classification

Assume that an emergency occurs at time t_1 and machine M_2 breaks down, and maintenance is expected to be completed at time t_2. Operation O_{21}, O_{12} and O_{22} will be directly affected by machine failure and cannot continue production, operation O_{33} will be indirectly affected, and other tasks will not be affected by machine failure. Therefore, J_1 and J_2 are directly affected tasks, J_3 are indirectly affected tasks, and the remaining tasks are unaffected tasks.

In the scheduling strategy based on task classification, the task completion degree should be determined when dynamic events occur. Then, different scheduling policies are adopted according to different types of dynamic events. Detailed procedures are described in the next section.

3.2. Task scheduling strategy

In the scheduling strategy based on task classification, when a dynamic event occurs, the task completion degree needs to be judged. Then, different scheduling strategies are adopted according to different types of dynamic events, and the detailed process is as follows:

Task completion degree determination: Determine whether the operation is complete according to the completion time End_{ij} of the static scheduling scheme and the fault time interval of the machine. On the device without fault, if $Start_{ij} < Break_e$, the j-th operation O_{ij} of task J_i is marked as processed; otherwise, it is classified as unprocessed operation and included in the set of tasks to be processed. On the faulty device, if $End_{ij} < Break_e$, the j th operation O_{ij} of J_i is marked as processed. Otherwise, the operation O_{ij} is marked as unprocessed and is waiting for reallocation.

Classification of unprocessed operations: According to the dynamic time type, the operations that have been marked as unprocessed are further subdivided into three categories according to the degree to which the task is affected, namely direct impact, indirect impact, and other.

(1) Event 1 Emergency order Arrives:

If the operation O_{ij} satisfies $P_{ij} < P_{i'j'}$, job J_i is determined to be the directly affected task. $P_{i'j'}$ indicates the priority of the urgent task.

If the operation O_{ij} satisfies $P_{ij} = P_{i'j'}$, job J_i is determined to be the task indirectly affected.

If the operation O_{ij} satisfies $P_{ij} > P_{i'j'}$ or $Start_{ij} < Break \ \&\& \ End_{ij} > Break$, the job J_i is determined to be the task indirectly affected.

(2) Event 2 Priority adjustment of unfinished tasks

If the operation O_{ij} satisfies $P_{ij} < P_{i'j'}$, the task is determined to be directly affected.

If the operation O_{ij} satisfies $P_{ij} \ge P_{i'j'}$ or $Start_{ij} < Break \ \&\& \ End_{ij} > Break$, the job J_i is determined to be unaffected.

(3) Event 3 Mechanical breakdown

Tasks on the failure of equipment, if the operation O_{ij} content type (4), the decision task J_i is directly affected task. Among them, the $Break_e$ moment of the failure of said equipment e , REP_e said equipment is after repair time.

$$\begin{gathered} Start_{ij} \le Break_e \ \&\&End_{ij} < Break_e + REP_e \\ or \\ Start_{ij} < Break_e \ \&\&End_{ij} > Break_e + REP_e \\ or \\ Start_{ij} > Break_e \ \&\&End_{ij} < Break_e + REP_e \end{gathered} \tag{4}$$

The task on the failure of equipment, if operation O_{ij} content type $Start_{ij} \ge Break_e + REP_e$, which task J_i for the task of indirectly affected. If in not the failure of equipment, operating O_{ij} content type $Start_{ij} \ge Break_e$, which task J_i is not affected by the equipment failure, as other types.

Task adjustment strategy:

Since different types of tasks may be affected differently by different dynamic events, different scheduling strategies are adopted for divided types of tasks in the rescheduling process to mitigate the impact of machine failures.

(1) Directly affected tasks

Directly affected tasks include two types of processes, unprocessed processes and partially completed processes (tasks include multiple processes and only some of them are processed). The remaining unprocessed processes still retain task flexibility and equipment selection flexibility.

(2) Indirect impact tasks

The flexibility of equipment selection is retained for indirectly affected tasks, and the processing sequence of the same priority is consistent with the initial scheduling scheme.

(3) Other tasks

The processing sequence of other tasks and equipment selection should be consistent with the initial scheduling scheme.

3.3. The crossover and mutation operations

In order to better express the strong process constraints in genes, this paper adopts the double integer coding mode. To improve the search efficiency of the algorithm and reduce the damage to the dominant solution, this paper uses two crossover operations and two mutation operations. The detailed steps are as follows.

(1) The operation crossover (POX)

All jobs were randomly divided into two sets A and B, and two individuals were randomly selected as parent individuals P1 and P2. The genes belonging to set A in P1 and P2 are put into the corresponding genes of offspring C1 and C2. Fill the genes belonging to set B to the free positions of C2 and C1 from left to right. POX is shown in Figure 3.

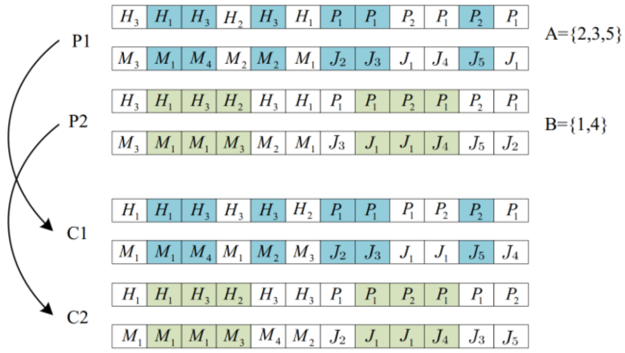


Fig. 3 POX process diagram

(2) Multi-point crossover (MPX)

A binary sequence of length was randomly generated, and two individuals were randomly selected as parent individuals P1 and P2. Put the gene of P1 corresponding to the index of 1 in the binary sequence into the gene position corresponding to the offspring C1. The gene of P2 corresponding to the index of 1 in the binary sequence is placed in the position of the gene corresponding to the offspring C2. Corresponding to the index P2 and P1 gene positions with the value of 0 in the binary sequence, the idle gene positions of C1 and C2 are filled in turn from left to right.

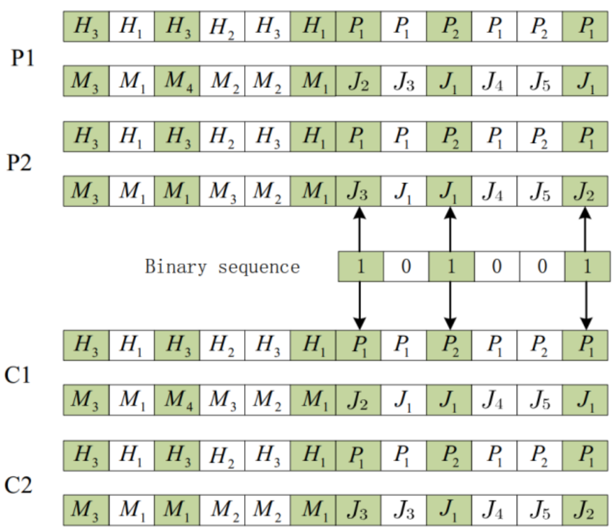


Fig. 4 MPX process diagram

(3) Mutation Operation based on task swapping

Two task exchange points are randomly generated to exchange the positions and corresponding information of two tasks.

(4) Mutation Operation based on task swapping

A random device is selected to select gene bits, and an integer is randomly generated to replace the original value.

Since the crossover and mutation operation may cause the device selection decoding error, the calculation method of

Equation (5) is adopted to decode in the decoding process.

$$M_{ij} = k \ mob \ M'_{ij} \tag{5}$$

The M_{ij} for the operation j of the task i corresponding equipment corresponding indexes, we can choose the equipment selection k as the value of the equipment selection genes, M'_{ij} is the optional equipment number of O_{ij}.

3.4. Individual density optimization strategy

Due to the random search of NSGAII algorithm, there are many similar individuals with the same characteristics in the population, which delays the optimization process of the algorithm. Inspired by Wen et al. [9], the individual density optimization strategy is adopted in this paper. By evaluating the characteristics of each individual, individuals with higher aggregation density in the same fitness value are removed, so that each layer of the population and the whole population maintain a certain degree of diversity, so that more individuals can participate in the evolution of the population.

3.5. Variable neighborhood Search Strategy

Variable neighborhood search algorithm has efficient local search ability and fast iteration speed. In this paper, the neighborhood search structure proposed by Li et al. [10] is used to adjust the key blocks of the scheduling results, and the scheduling area is divided into three types. As shown in Figure 5, in the first critical region 1, only the first two specific orders of operations are swapped. In region 2, the operations of the head and its neighbors and the operations with the tail and its neighbors are swapped separately, while the exchange of internal operations is avoided. In region 3, only the two adjacent operations located at the tail of the region are exchanged. If there is only one operation in the critical block, no transformation is performed. In the figure, the moving process of the neighborhood is $\{(O_{11} \to O_{12}),(O_{32} \to O_{13}),(O_{33} \to O_{25}),(O_{22} \to O_{23})\}$.

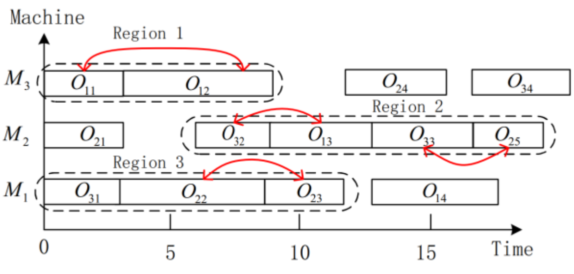


Fig. 5 Exchang critical operations block

3.6. Overall framework of the algorithm

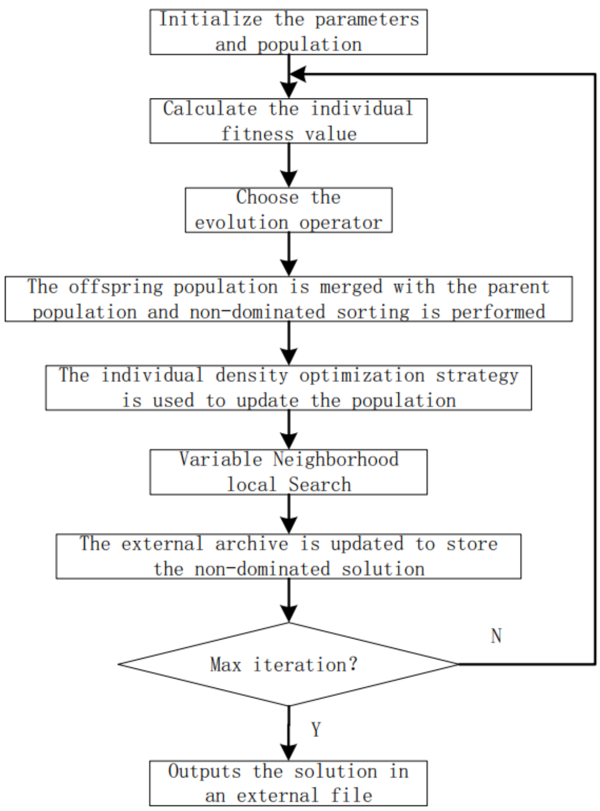


Fig. 6 MNV algorithm flow chart

4. Experimental verification and analysis

4.1. Experimental Setup

The experiments were run on a device with an Intel i7-8700 CPU @3.2GHZ processor, Windows 10 OS, and 8GB memory. All the experimental codes were implemented on Python 3.8. The initial schedule is solved using PSO. The experiment consists of two parts to validate the proposed algorithm and the dynamic scheduling policy. The first part of the experiment discusses the applicability of the process adjustment method to solve the FJSP problem with machine failures. The second part of the experiment verifies the effectiveness of MNV in dealing with the mechanical fault FJSP problem. The parameters of the algorithm are shown in Table 1.

Table 1. Parameter configuration

Name	Parameter
Population size	300
Probability of crossover	0.8
Mutation probability	0.2
Maximum number of iterations	500

4.2. Comparative experimen

In order to verify the effectiveness of the rescheduling strategy of the proposed MNV algorithm in the production shop, this experiment constructs flexible FJSP instance data with strong process constraints for testing on the basis of 20 sets of FJSP problem benchmarks. In order to maintain the objective fairness of the experiment, all the comparison experiments were carried out in the same scenario, and the test scenario is described in detail below.

A set of fault parameters were randomly generated as the experimental scenario, the number of faulty machines was randomly selected according to the total number of equipment (1-10%* the total number of machines), and the faulty equipment was randomly specified, and the mean value of the equipment failure time and the equipment maintenance time should meet the Poisson distribution. In order to compare the impact of the repair time on the scheduling strategy, the repair time is set to multiple levels, which are 100 minutes, 150 minutes and 200 minutes.

The MNV algorithm and the designed three task adjustment strategies are applied to the dynamic scheduling problem of building machine failures to verify the superiority and applicability of the MNV algorithm process adjustment strategy. The details of the strategy are as follows:

$$\textit{Pareto dominance ratio} = \frac{\text{Total number of solutions}}{\text{The total number of solutions obtained by all strategies}} \times 100\%. \tag{6}$$

Table 2. The number of non-dominated solutions obtained by four strategies.

Time of maintenance	The problem	Total	Strategy1	Strategy2	Strategy3	MNV	Ratio of dominance
100	1	1	0	0	0	**1**	100.0
	2	30	12	0	2	12	40.0
	3	5	0	1	1	**4**	80.0
	4	16	0	0	0	**15**	93.8
	5	8	1	0	0	**8**	100.0
	6	16	0	0	2	**14**	87.5
	7	19	0	14	0	**19**	100.0
	8	14	0	0	3	**11**	78.6
	9	1	0	1	0	1	100.0
	10	19	2	0	14	**18**	94.7
	11	34	0	10	0	**33**	94.1
	12	24	10	0	1	23	91.3
	13	26	14	12	4	**26**	100.0
	14	29	4	0	0	**27**	93.1
	15	40	0	0	8	**34**	85.0
	16	26	3	0	0	**19**	73.1
	17	65	5	20	1	**48**	73.8
	18	44	12	0	0	**24**	54.5
	19	34	0	4	1	**19**	55.9
	20	10	6	0	0	4	40.0
150	1	10	1	0	5	**10**	100.0
	2	35	0	7	0	**23**	65.7
	3	24	21	0	0	21	87.5
	4	17	0	0	4	**17**	100.0
	5	6	3	0	0	**6**	100.0
	6	11	1	4	1	4	36.4
	7	14	5	9	0	**10**	71.4
	8	2	0	0	2	2	100.0
	9	45	0	1	0	**36**	80.0
	10	23	0	5	18	**23**	100.0
	11	14	5	0	0	**14**	100.0
	12	53	0	34	0	**52**	98.1
	13	37	1	0	0	**34**	91.9
	14	70	0	0	1	**60**	85.7
	15	69	50	7	12	**64**	92.8
	16	90	9	14	0	**60**	66.7
	17	34	1	0	19	**34**	100.0
	18	27	4	0	0	**20**	74.1
	19	24	0	2	0	**24**	100.0
	20	50	0	0	0	**32**	64.0
200	1	12	0	0	0	**12**	100.0
	2	30	1	0	5	**28**	93.3
	3	34	5	0	0	**33**	97.1
	4	18	0	1	0	**18**	100.0
	5	2	0	0	1	**2**	100.0
	6	17	1	0	2	**5**	29.4
	7	1	0	1	0	1	100.0
	8	21	1	0	0	**12**	57.1
	9	24	0	0	14	13	54.2
	10	64	10	36	8	**53**	82.8
	11	14	0	1	0	**10**	71.4
	12	28	0	0	0	**26**	92.9
	13	34	1	0	0	**28**	82.4
	14	29	9	0	1	**29**	100.0
	15	47	0	5	0	**41**	87.2
	16	59	0	0	0	**50**	84.7

Continued 2

Time of maintenance	The problem	Total	Strategy1	Strategy2	Strategy3	MNV	Ratio of dominance
200	17	64	1	0	0	53	82.8
	18	53	5	21	18	39	73.6
	19	28	0	0	1	28	100.0
	20	41	0	4	0	39	95.1

(1) Strategy 1

The task status is not marked for all tasks, and the task retains the original production equipment selection flexibility.

(2) Strategy 2

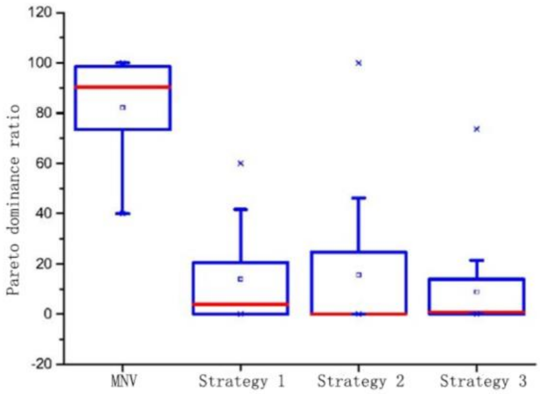
Mark the task status for the unfinished tasks. The processing machines that directly affect the task are flexible and optional, but the processing machines for other jobs are fixed and unchanged.

(3) Strategy 3

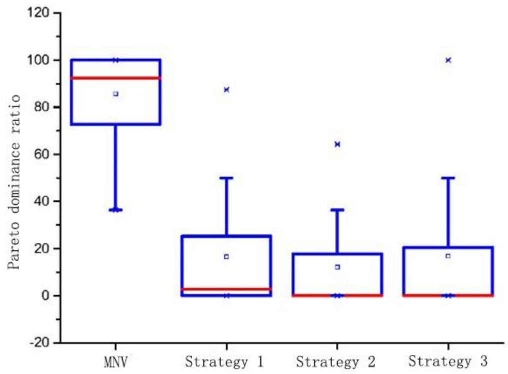
The task status is marked for the unfinished tasks, and the processors that directly affect the job and indirectly affect the job are flexible and optional, but the processors for other jobs are fixed.

The test results record the number of pareto solutions for each set of problems, as shown in Tables 2. Among them, the bold font in the table indicates that the proposed algorithm outperforms other strategies, and the pareto dominance ratio in the last column of the table is calculated by equations (6).

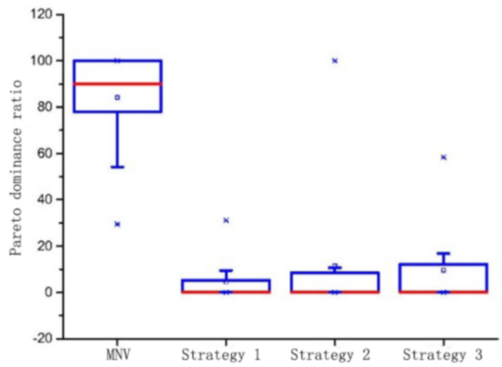
According to the data analysis in Table 2, when the maintenance time of the faulty equipment is set to 100 minutes, the MNV algorithm finds the largest number of pareto solutions on 16 sets of problems, among which the pareto dominance rate in a total of 18 sets of problems is more than 50%. When the repair time is set to 150 minutes, the MNV algorithm outperforms the other strategies on a total of 17 sets of problems, and there are 19 sets of problems with pareto dominance above 50%. With a repair time of 200 minutes, the MNV strategy outperforms the other strategies on 18 sets of problems and achieves pareto dominance above 50% on 19 sets of problems. The corresponding box plots of the dominance rates of the four strategies are shown in Figure 7. It is worth noting that strategy 2 and strategy 3 include the proposed scheduling mechanism based on task classification, and their various indicators are better than strategy 1, which can prove the rationality and effectiveness of the MNV algorithm strategy.



(a) The failure maintenance time is 100 hours



(b) The failure maintenance time is 150 hours



(c) The failure maintenance time is 200 hours

Fig. 7 Non-dominated unpacking diagram of four strategies

5. Summary

In this paper, the dynamic job-shop scheduling problem is studied from two aspects of dynamic scheduling strategy and algorithm, and a scheduling method based on improved NSGAII algorithm (MNV algorithm) is proposed. Compared with the existing dynamic scheduling algorithms, the design of MNV algorithm not only improves the stability of the rescheduling scheme, but also improves the search efficiency of the algorithm. Finally, benchmark problems are used to validate the proposed method. The results show that the MNV algorithm can effectively solve the dynamic scheduling problem.

References

[1] Oluyisola O E, Bhalla S, Sgarbossa F, et al. Designing and developing smart production planning and control systems in the industry 4.0 era: a methodology and case study[J]. Journal of Intelligent Manufacturing, 2022, 33(1): 311-332.

[2] Wang G G, Gao D, Pedrycz W. Solving multiobjective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm[J]. IEEE Transactions on Industrial Informatics, 2022, 18(12): 8519-8528.

[3] Pan Y, Gao K, Li Z, et al. Solving biobjective distributed flow-shop scheduling problems with lot-streaming using an improved Jaya algorithm[J]. IEEE Transactions on Cybernetics, 2022.

[4] Zhang F, Mei Y, Nguyen S, et al. Evolving scheduling heuristics via genetic programming with feature selection in dynamic flexible job-shop scheduling[J]. IEEE Transactions on Cybernetics, 2021, 51(4): 1797-1811.

[5] An Y, Chen X, Gao K, et al. Multiobjective flexible job-shop rescheduling with new job insertion and machine preventive maintenance[J]. IEEE Transactions on Cybernetics, 2022.

[6] Zhang Y, Zhu H, Tang D, et al. Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems[J]. Robotics and Computer-Integrated Manufacturing, 2022, 78: 102412.

[7] Wen X, Lian X, Qian Y, et al. Dynamic scheduling method for integrated process planning and scheduling problem with machine fault[J]. Robotics and Computer-Integrated Manufacturing, 2022, 77: 102334.

[8] Zhang F, Mei Y, Nguyen S, et al. Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling[J]. IEEE Transactions on Evolutionary Computation, 2021, 25(4): 651-665.

[9] Wen X, Lian X, Qian Y, et al. Dynamic scheduling method for integrated process planning and scheduling problem with machine fault[J]. Robotics and Computer-Integrated Manufacturing, 2022, 77: 102334.

[10] Li X, Gao L, Pan Q, et al. An Effective Hybrid Genetic Algorithm and Variable Neighborhood Search for Integrated Process Planning and Scheduling in a Packaging Machine Workshop [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 49(10):1933-1945.