TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/24 pp1390–1408
DOI: 10.26599/TST.2023.9010141
Volume 29, Number 5, October 2024

Flexible Job Shop Composite Dispatching Rule Mining Approach Based on an Improved Genetic Programming Algorithm

Xixing Li, Qingqing Zhao, Hongtao Tang*, Xing Guo, Mengzhen Zhuang, Yibing Li, and Xi Vincent Wang

Abstract: To obtain a suitable scheduling scheme in an effective time range, the minimum completion time is taken as the objective of Flexible Job Shop scheduling Problems (FJSP) with different scales, and Composite Dispatching Rules (CDRs) are applied to generate feasible solutions. Firstly, the binary tree coding method is adopted, and the constructed function set is normalized. Secondly, a CDR mining approach based on an Improved Genetic Programming Algorithm (IGPA) is designed. Two population initialization methods are introduced to enrich the initial population, and a superior and inferior population separation strategy is designed to improve the global search ability of the algorithm. At the same time, two individual mutation methods are introduced to improve the algorithm's local search ability, to achieve the balance between global search and local search. In addition, the effectiveness of the IGPA and the superiority of CDRs are verified through comparative analysis. Finally, Deep Reinforcement Learning (DRL) is employed to solve the FJSP by incorporating the CDRs as the action set, the selection times are counted to further verify the superiority of CDRs.

Key words: flexible job shop scheduling; composite dispatching rule; improved genetic programming algorithm; deep reinforcement learning

1 Introduction

In recent years, the global economy has experienced rapid development, leading to the reform and innovation of job shop models. The traditional large-scale manufacturing has gradually changed to small batch and multi variety customization, and meanwhile, the energy utilization and processing efficiency have been paid more and more attention by complex product manufacturing enterprise, such as the special vehicle equipment manufacturing enterprises. The special vehicle plays an important role in special task scenarios such as patrol, transportation, disaster prevention, etc. that can effectively improve the efficiency of social development with their specialized nature. With the diversification of market demand, special vehicle manufacturing enterprises also face many challenges, and how to quickly deliver products has become the core of enterprise competition. In the practical manufacturing workshop, each part can be processed by different manufacturing equipment, and the processing time of the same type of parts also varies with the different working years, manufacturers and power consumption of the equipment. At the same

• Xixing Li, Qingqing Zhao, and Xing Guo are with Hubei Key Laboratory of Modern Manufacturing and Quality Engineering, School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China. E-mail: li_xi_xing@126.com; zqq410327@163.com; gx_study2022@163.com.
• Hongtao Tang, Mengzhen Zhuang, and Yibing Li are with Hubei Digital Manufacturing Key Laboratory, School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China. E-mail: tanghongtaozc@163.com; zmz990910wy@163.com; ahlyb@whut.edu.cn.
• Xi Vincent Wang is with Department of Production Engineering, KTH Royal Institute of Technology Stockholm SE-10044, Sweden.
* To whom correspondence should be addressed.
Manuscript received: 2023-07-19; revised: 2023-11-11; accepted: 2023-11-14

time, with the increase of uncertain interference factors in practical production environment, the traditional scheduling strategy cannot meet the demand of scheduling manager. Therefore, how to generate feasible and effective scheduling scheme is a key problem that managers urgently need to solve. The scheduling of the special vehicle manufacturing workshop can be taken as the Flexible Job shop Scheduling Problem (FJSP), which is an extension of the Job Shop Problem (JSP) and widely recognized as a NP-hard problem[1]. To address this challenge, intelligent search algorithms are often employed to obtain better approximate optimal solutions[2-4]. The intelligent search algorithm provides a specific approximate optimal solution for the FJSP[5]. Meng et al.[6] addressed the Autonomous Guided Vehicles (AGV) constraint problem in the FJSP and propose an improved Genetic Algorithm (GA) to solve it. Long et al.[7] combined the Q-learning algorithm with the artificial swarm algorithm to create a self-learning artificial swarm algorithm for the insertion of new jobs in the FJSP and demonstrate the effectiveness of the algorithm through example analysis. Li et al.[8] applied a Hybrid Chemical Reaction Optimization (HCRO) algorithm to address the distributed FJSP, they develop a novel encoding-decoding method for Flexible Manufacturing Unit (FMU) and design an improved method of critical-FMU to improve the global and local search ability of the algorithm. Duan and Wang[9] introduce two metrics to assess the robustness of the system which refers to dynamic events like machine failures and new job arrivals, a multi-objective particle swarm optimization algorithm has been constructed to solve it that mainly focuses on the total system energy consumption, manufacturing time, and integrated reusability. The improved intelligent search algorithm can yield better results when solving the FJSP. However, the time required to solve the problem increases as the problem size increases[10], and it needs to be solved again when the shop processing parameters change.

With the scale of the problem increasing, it not only results in higher computational cost but also fails to provide a more desirable approximate optimal solution within a reasonable time to meet the scheduling requirements. Therefore, the Dispatching Rule (DR)[11,12] which can solve the FJSP with a lower computational complexity has been attracted by many research scholars. The DR method has a lower computational complexity and is easier to implement, resulting in faster response speeds. It can effectively meet the actual production needs of the workshop. When compared to heuristic algorithms, the advantages of the DR method are more pronounced for large-scale workshop scheduling problems[13]. DR, known as the priority assignment rule[14], sorts the processed workpieces based on their priority and addresses two subproblems in shop scheduling: operations sequencing and machine selection[15]. The comparison between the intelligent search algorithm and DR for solving FJSP is shown in Table 1.

The DR can be separated into single DR (SDR) and composite DR (CDR). The SDR contains the Earliest Due Date (EDD)[16], Shortest Processing Time (SPT)[17], First In First Out (FIFO)[18], Longest Processing Time (LPT), Slack Time (ST), etc.[19] Chen and Matis[20] applied an SDR to optimize the mean tardiness of weighted workpieces in a job shop and achieve favorable outcomes. Mihoubi et al.[21] proposed an agent-assisted simulation-based experimental optimization method for the combined optimization problem associated with FJSP, it can balance the short-term reactivity of the shop in the face of repeated perturbations and the overall performance of the manufacturing system. Heger and Voss[22] integrated the traditional FJSP with AGV to optimize the operation sequence and vehicle scheduling. Teymourifar et al.[23] constructed a due date allocation model of a dynamic FJSP, propose a new SDR based on the combination characteristics of job shops, and validate the model through simulation. Đurasević and Jakobović[24] proposed a DR selection procedure to select the appropriate DR from the set of evolved DRs based on the features of the problem instances to be solved, which achieves better results than those

Table 1 Comparison of intelligent search algorithms and DR.

Contrast content	Intelligent search algorithm	DR
Calculation result	Specific solution	Priority rule
Scheduling method	Specific solution guides scheduling program	Priority guides scheduling program
Complexity	Relatively high	Relatively low
Reusability	Not reusable	Reusable

obtained when only a single DR is selected. Now, some literatures use deep reinforcement learning (DRL) to solve such problems[25]. Chang et al.[26] focused on providing an optimal dispatching algorithm for the AGV in a mobile metal board manufacturing facility, and used DRL to learn an AGV's dispatching algorithms. Song et al.[27] considered the well-known FJSP and address these issues by proposing a novel DRL method to learn high-quality Priority Dispatching Rules (PDRs) end to end. Liu et al.[28] developed a novel DRL-based PMS method, this can efficiently select a sequence of DRs based on the current environment or unexpected events. Gui et al. [29] proposed a scheduling method based on DRL for the dynamic flexible job-shop scheduling problem, which achieves significantly better scheduling results than an SDR and the DQN-based method. Min and Kim[30] proposed a reinforcement learning-based Apparent Tardiness Cost (ATC) DR that estimates the lookahead parameter directly from raw job data. The proposed DR displayed the best performance in the main experiment, which compared the performance with five existing DRs. However, the applications of SDR only utilize a portion of the feature information from the shop production scheduling process, resulting in suboptimal performance[31]. To overcome the limitation of the SDR, the CDR is obtained by combining the SDR with different operators. The CDR has higher performance and inherits the advantages of SDR [32]. There are two methods for establishing CDR: manual method, and heuristic combination by using intelligent algorithm. Because the characteristic parameters of FJSP are generally large, designing CDR manually is not only time-consuming but also leads to relatively poor optimization performance[33, 34]. With the advancement of intelligent algorithms, several researchers have explored the generation of CDR by applying the intelligent algorithms. Durasević and Jakobović[35] employed an evolutionary algorithm to generate job shop CDR and achieve improved mean flow time. Shady et al.[36] developed an adaptive feature selection mechanism that utilizes a new expression for Genetic Programming (GP) rules. The GP algorithm is relatively simple to operate and offers high flexibility and a learning mechanism, which is suitable for heuristic combination of different SDRs[37]. Xu et al.[38] developed a heuristic template for making delayed routing decisions and three different delayed routing strategies with optimization objectives of energy efficiency and mean tardiness, the results of compared experiments demonstrate that the Genetic Programming Hyper-Heuristic (GPHH) approach with delayed routing outperforms the state-of-the-art GPHH approach. Zhang et al.[39] proposed a hybrid GP algorithm to develop a DR for an intelligent job shop, and verify the effectiveness of the developed algorithm by conducting a study on two real scenarios. Jun and Lee[40] introduced a decision-tree-based approach with feature construction and tree-based learning, it can automatically extract DR from existing or well-performing DRs. Wang et al.[41] conducted research on the daily surgery scheduling problem with operating room eligibility and dedicated surgeons, and propose an adaptive composite dispatching method that combines three popular rules to address this problem. Nguyen et al.[42] developed a new agent-assisted GP algorithm to improve the computational requirements, accuracy, and quality of the evolutionary rules of the GP algorithm. Mei et al.[43] proposed a GP algorithm-based feature selection algorithm for FJSP which can effectively reduce the computation time, and the algorithm's fitness values are evaluated by using a simulation model.

The GP algorithm, commonly used for mining CDR in workshops, has shown high performance. However, few studies have applied the Improved GP Algorithm (IGPA) to solve FJSP, and also few studies have used DRs as the action set in the computation process of DRL-based FJSP solution that can verify the different performance of CDR and SDR by counting the selected times. Therefore, this paper presents an IGPA for mining CDR in solving FJSP. Firstly, a strategy of separating the population into superior and inferior populations are introduced to increase diversity and avoid early convergence to local optimal solutions. Secondly, a leaf node mutation and non-leaf node mutation strategy are applied to enhance the algorithm's local search ability. Additionally, a normalization method is employed to normalize the terminator set of IGPA individuals, ensuring the algorithm's overall performance. Moreover, the optimized CDR's performance is validated by applying it to solve FJSP with different sizes and comparing it with existing SDR. Finally, the DRL is used to solve FJSP to further demonstrate CDR' superiority by taking different DRs as the action set of DRL.

The remainder of this paper is divided into five sections, Section 2 describes the FJSP and corresponding model construction. Section 3 introduces the proposed IGPA. Sections 4 and 5 present the simulation experiments, comparisons and analysis. Section 6 shows the conclusions and future work.

2 Problem Description and Model Construction

FJSP can be described as follows: In FJSP, there are n workpieces to be processed, denoted as $J=\{J_1, J_2,...,J_n\}$, which can be processed on m machines denoted as $M=\{M_1,M_2,...,M_m\}$. Each workpiece J_i has N_i operations with sequential constraints, expressed as $O_i=\{O_{i,1},O_{i,2},...,O_{i,N_i}\}$. Each operation O_{ij} has one or more optional processing machines, denoted as $M_{i,j}\subseteq M$. Additionally, each operation alternative concentration of machines to process the operation consumes different processing times.

2.1 Assumptions

To facilitate the construction of the mathematical model, the following assumptions are introduced in this paper for FJSP:

(1) All equipment is available at time 0.

(2) Once a workpiece has begun processing, it cannot be interrupted until it is completed.

(3) The conversion time of the processing equipment and transportation time between different operations are not considered.

(4) Machine failure, material shortage, etc. are not considered.

(5) The processes of different workpieces are independent of each other and there is no sequential constraint relationship.

(6) The due date for each workpiece is predetermined and fixed during processing.

2.2 Variable definition

Table 2 Parameter and meaning.

Parameter	Meaning
n	Total number of workpieces
m	Total number of machines
N_i	Total number of operations for workpiece i
i	Workpiece index number, $i\in\{1,2,...,n\}$
j	Work order index number, $j\in\{1,2,...,N_i\}$
k	Machine index number, $k\in\{1,2,...,m\}$
l	The index number of the processing machine of the j-th operation

(to be continued)

Table 2 Parameter and meaning. (continued)

Parameter	Meaning
J_i	The i-th workpiece to be processed
$O_{i,j}$	The j-th operation of J_i
M_k	The k-th machine
$M_{i,j}$	The set of optional machines for J_i in operation j
C_i	Completion time of the last operation of the workpiece i
D_i	Due date of J_i
MN_j	Number of machines available for the j-th operation
$M_{j,l}$	The l-th available machine of the j-th operation
$S_{i,j}$	Start processing time of operation $O_{i,j}$
P_{ijk}	Processing time of operation $O_{i,j}$ on the machine M_k
I	A large enough positive number
b_i	Total number of operations of workpiece J_i
$E_{i,j}$	Completion time of the operation j corresponding to the workpiece i
$\theta_{i,j}^{i',j'}$	0-1 variable: if operation $O_{i,j}$ is processed before operation $O_{i'j'}$, $\theta_{i,j}^{i',j'}$ =1; otherwise $\theta_{i,j}^{i',j'}$ =0
α_{ijl}	0-1 variable: if operation $O_{i,j}$ is processed on machine $M_{j,l}$, α_{ijl}=1; otherwise α_{ijl}=0
β_{ijk}	0-1 variable: if operation $O_{i,j}$ is selected to machine M_k, β_{ijk}=1; otherwise β_{ijk}=0

2.3 Model construction

Job shop scheduling aims to optimize one or more specific performance indicators by arranging operations sequencing and machine selection while meeting assumptions and constraints. In this paper, the objective of the FJSP is to minimize the MCT.

Objective function:

$$f_1=\max\{C_i|i=1,2,...,n\} \tag{1}$$

The constraint in Eq. (2) states that each operation of each workpiece can only be processed by one optional machine.

$$\sum_{l=1}^{MN_j}\alpha_{ijl}=1,\forall i,j;i=1,2,...n;j=1,2,...,b;l=1,2,...,MN_j \tag{2}$$

The constraint in Eq. (3) indicates that a machine can only process one operation of a workpiece at a time, and there are sequential constraints for different workpieces on the same processing machine.

$$S_{i,j}+\sum_{k=1}^{m}P_{ijk}\beta_{ijk}\leqslant S_{i',j'}+I(1-\theta_{i,j}^{i',j'}),i,i'=1,2,...n;$$
$$j=1,2,...,b_i;j'=1,2,...,b_{i'} \tag{3}$$

Each workpiece must follow a specific process route and there is a sequence relationship between the various operations of the same workpiece, as expressed in Eq. (4) and Eq. (5).

$$E_{i,j} \leqslant S_{i',j'} + I\cdot(1-\theta_{i,j}^{i',j'}), \forall i,i',j,j'; i,i' = 1,2,\ldots,n;$$
$$j,j' = 1,2,\ldots b_i \tag{4}$$

$$E_{i',j'} \leqslant S_{i,j} + I\cdot\theta_{i,j}^{i',j'}, \forall i,i',j,j'; i,i' = 1,2,\ldots,n;$$
$$j,j' = 1,2,\ldots b_i \tag{5}$$

The constraint in Eq. (6) indicates that once processing has begun, the workpiece cannot be interrupted.

$$E_{i,j} = S_{i,j} + \sum_{k=1}^{m} P_{ijk}\beta_{ijk}, i = 1,2,\ldots,n; j = 1,2,\ldots,b_i \tag{6}$$

2.4 Composition of combination rule

CDR is a heuristic approach that improves production scheduling processes by combining information from multiple sources. It is typically represented by a binary tree, which includes a terminator set and a function set. These sets contain important information for the algorithm's operation. Therefore, selecting the appropriate function set and terminator set not only reduces the search space of the algorithm but also enhances the rationality of CDR, which is crucial for improving the effectiveness of the solution.

(1) Determination of terminator set

The representative characteristic parameters of the workshop are selected as the terminator set of CDR to obtain higher-quality solutions for solving FJSP, and the specific terminator sets are shown in Table 3.

(2) Determination of function set

Based on Reference [37], in order to get a better combination of DRs and ensure population diversity, the maximum and minimum functions are introduced and the specific function set is shown in Table 4.

Table 3 Terminator set.

Full name	Abbreviation	Meaning
Processing Time	PT	Processing time of operation
Weight	W	Workpiece weight
Due Date	DD	Due date of the workpiece
Num of Operations	NOP	Total number of operations of workpiece
Avg Total Process Time	ATPT	Average total processing time of the workpiece
Remaining Processing Time	RPT	Remaining processing time of the workpiece
Num of Operations Remaining	NOR	Number of operations remaining on the workpiece
Current Time	CT	Current time
Next Processing Time	NPT	Processing time of next operation
Release Date	RD	Workpiece Release date

Table 4 Function set.

Function symbol	Meaning
+	Terminator summation
−	Terminator subtraction
*	Terminator multiplication
/	Divide by terminator and return 1 if the divisor is 0
Max	Take maximum value
Min	Take minimum value

3 IGPA

The GP algorithm, proposed by Koza in 1994, is an evolutionary computational method that imitates the idea of biological evolution[44]. Similar to GA, it contains population initialization, crossover, mutation, and selection for iterative operations to obtain an approximate optimal solution. However, the GP algorithm distinguishes itself by allowing dynamic changes in the logical structure and size of individuals, making it suitable for solving more complex problems[45]. Unlike heuristic optimization algorithms that produce specific solutions, the GP algorithm generates heuristic rules that can produce specific solutions. An example of a CDR generated by the GP algorithm is shown in Fig. 1, the terminator of the binary tree represents an SDR, and the expression of the CDR is obtained by performing an inorder traversal of the binary tree.

The CDR guides the FJSP in generating the pseudo-code process for a specific scheduling scheme as shown in Algorithm 1.

In the heuristic combination of SDRs, the terminator and function sets of the formed binary tree are randomly chosen. However, the non-uniformity of the

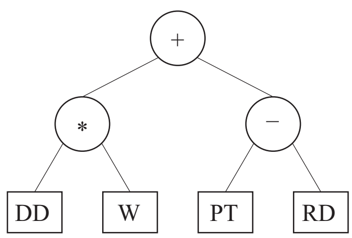


Fig. 1 GP tree of "(DD*W) + (PT−RD)".

Algorithm 1 CDR guidance generation scheduling program

```
Input Workpiece set J
Output Specific scheduling solution
Begin
    Calculation of the priority of the operation based on CDR;
    Ranking of operations according to their priority;
    If the sequence has the same priority do
        Randomizing the operations;
    Else
        Ranking of operations by priority;
    End if
    Selection of equipment for the process based operation sequence;
    Calculate the cumulative process time for the optional processing machine;
    If only one machine with the shortest cumulative process time do
        Select this equipment to process the operation;
    Else
        Random selection of processing equipment;
    End if
    Return Operation sequence and machine selection;
End
```

terminator's magnitude often leads to more meaningless solutions when direct function operations are performed. This significantly hampers the evolution speed of the algorithm. For instance, the generated DR (DD*W) + (PT−RD) cannot be functionally operated theoretically. Nevertheless, it is inevitable to generate such solutions during the population evolution process. To improve the overall performance of the algorithm for its normalization process, this paper employs a normalization treatment for an SDR, the priority value of a SDR is unified in the range of 0-1 to facilitate the calculation of the algorithm, and it is applicable to large, medium, and small three scale models, it is shown in Eq. (7).

$$R_{a,i}=\frac{a_i-\alpha_{\min}}{\alpha_{\max}-\alpha_{\min}} \tag{7}$$

In Eq. (7), $\alpha_{\min}$ and $\alpha_{\max}$ are the minimum and maximum values of the workpiece attribute a determined by an SDR in the workpiece to be processed; a_i is the value of attribute a of workpiece i and $R_{a,i}$ is the priority of workpiece i determined by attribute a after the normalization process.

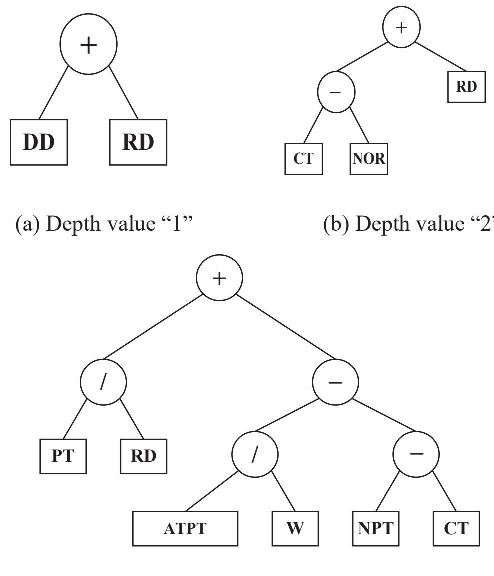
Taking the NOR as an example, the priority of the workpiece i after the normalization process can be expressed as

$$R_{NOR,i}=\frac{NOR_i-NOR_{\min}}{NOR_{\max}-NOR_{\min}} \tag{8}$$

In Eq. (8), $NOR_{\min}$ and $NOR_{\max}$ are the minimum and maximum values of the number of remaining operations of the workpieces to be processed, respectively; NOR_i is the number of remaining operations of the workpiece i; $R_{NOR,i}$ is the priority of the workpiece i determined by the SDR and NOR after normalization.

3.1 Population initialization

The genetic diversity of the offspring population is primarily influenced by the initial population. According to the method described in Ref. [40], half of the individuals are generated using the growth generation method, while the other half are generated using the full generation method. The growth generation method randomly selects the depth value within a given maximum depth to create the IGPA tree, and the full generation method sets the depth of the IGPA tree to ensure that all created IGPA trees have the same depth and shape. Therefore, the hybrid generation method combines the advantages of them and plays a crucial role in speeding up algorithm convergence. Figures 2 and 3 show the individuals generated by the growth generation method and the full generation method, respectively. In Fig. 2, the maximum depth is set to "3", and individuals are constructed by randomly selecting depth values between "1" and "3" to form the IGPA tree. In Fig. 3,



(a) Depth value "1" (b) Depth value "2"

(c) Depth value "3"

Fig. 2 Growth generative IGP tree.

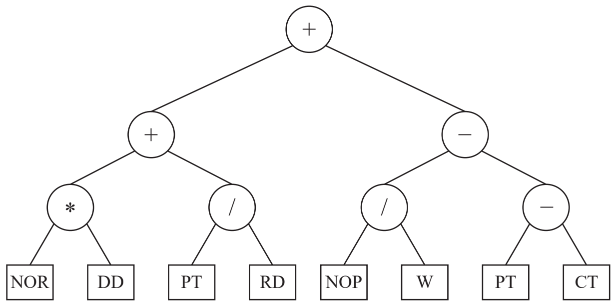


Fig. 3 Full generative IGP tree with value “3”.

the full generation method applies a fixed depth value “3”.

3.2 Adaptation evaluation

The IGPA individual refers to a collection of computer programs that are represented as a tree structure. These programs are capable of calculating the processing priority of a workpiece and determining various parameters such as completion time and delay time based on the priority. For this study, the MCT is selected as the adaptation function to address the scheduling problem.

3.3 Population evolution

(1) Crossover

In IGPA, two parent individuals are randomly chosen and a random number is generated. This number is then compared with the crossover probability. If the random number is smaller than the crossover probability, the crossover operation is executed. Otherwise, the operation proceeds to the next step. During the crossover operation, two non-leaf nodes are randomly selected from the parent generation. The subtrees with these two nodes as the root nodes are exchanged with each other, resulting in two child individuals. This process is shown in Fig. 4.

(2) Mutation

When performing mutation operations, randomly select nodes other than the root node as mutation points. There are two mutation forms: leaf nodes and non-leaf nodes mutation. If the selected node is a leaf node, select an element from the terminator set to replace it. If the selected node is a non-leaf node, deleting the subtree with this node as the root and then regenerating a new subtree. The depth of the new subtree should not exceed the maximum depth, as shown in Fig. 5.

(3) Selection

To increase the evolutionary rate of the population

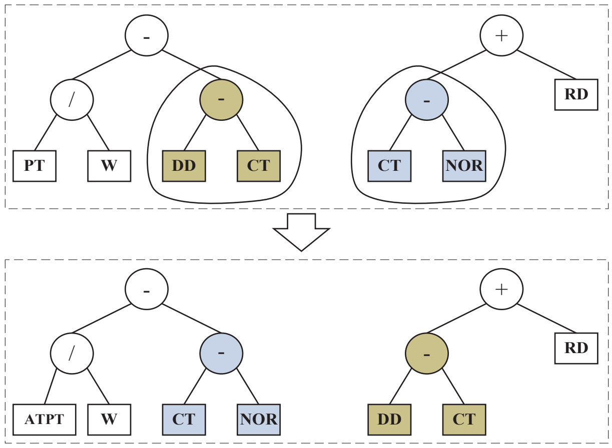


Fig. 4 IGPA tree crossover operation.

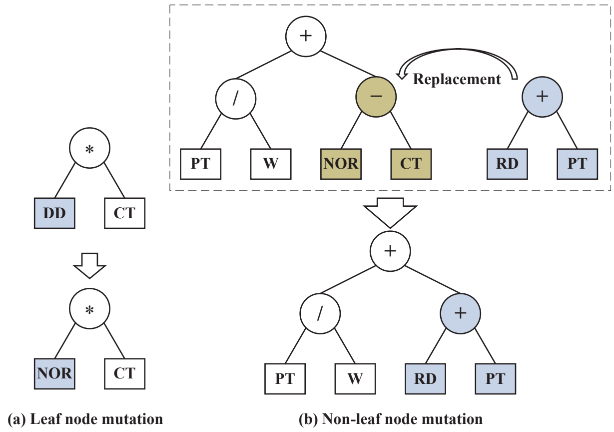


Fig. 5 IGPA tree mutation operation.

and maintain diversity, a roulette selection operation is introduced during the selection operation, as well as incorporating an elite retention strategy, and the individuals with the best fitness values in the previous generation are directly retained to the next generation. To prevent the algorithm failing into the local optimal solution, a dynamic population strategy is used to divide the population into superior population and inferior population according to the size of fitness value. To ensure the high-quality individuals in the population have a high probability of being inherited and improve the population diversity, each population size is set to 50% of the total population. Different update strategies are implemented for the two populations to improve the local and global search ability of the IGPA, which is shown in Fig. 6.

3.4 IGPA flowchart

The flowchart IGPA is illustrated in Fig. 7, and the execution steps are as follows:

Step 1: Generating the initial population by the

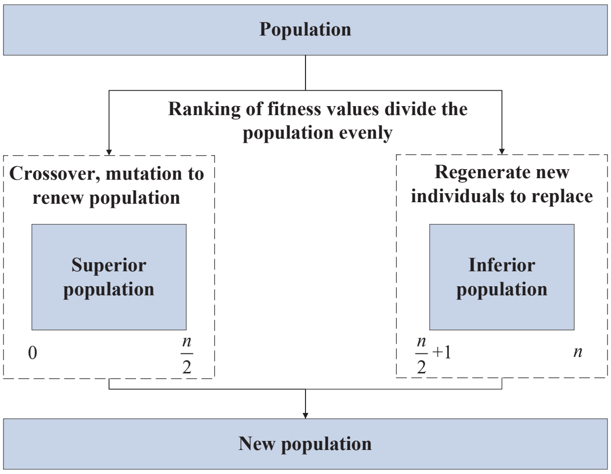


Fig. 6 Strategies for separating superior and inferior population.

population initialization strategy in Section 2.1.

Step 2: Calculating the population fitness value and ranking them in ascending order by fitness value, and dividing the ranked populations into two equal parts to get the "superior population" and "inferior population".

Step 3: Updating the "superior population" by applying the crossover and mutation strategies in Section 2.3, the "inferior population" is updated by randomly generating new individuals instead of the old ones.

Step 4: Incorporating the updated "superior population" and "inferior population" and the roulette selection operation and the elitist preservation strategy are used for population evolution.

Step 5: Repeating **Step 2** – **Step 4** until the algorithm satisfies the termination condition and outputting the solution.

4 Simulation Experiment

4.1 Simulation scenario construction

In order to verify the effectiveness of scheduling rules guiding the generation of scheduling schemes for shop scheduling problems, the FJSP is extended on the basis of special vehicle parts production workshops. Based on the resource selection constraints and the level of flexibility, the FJSP can be classified into two types: total FJSP (T-FJSP) and partial FJSP (P-FJSP). The degree of flexibility coefficient is introduced in this paper, and dividing the FJSP into three types: 100%

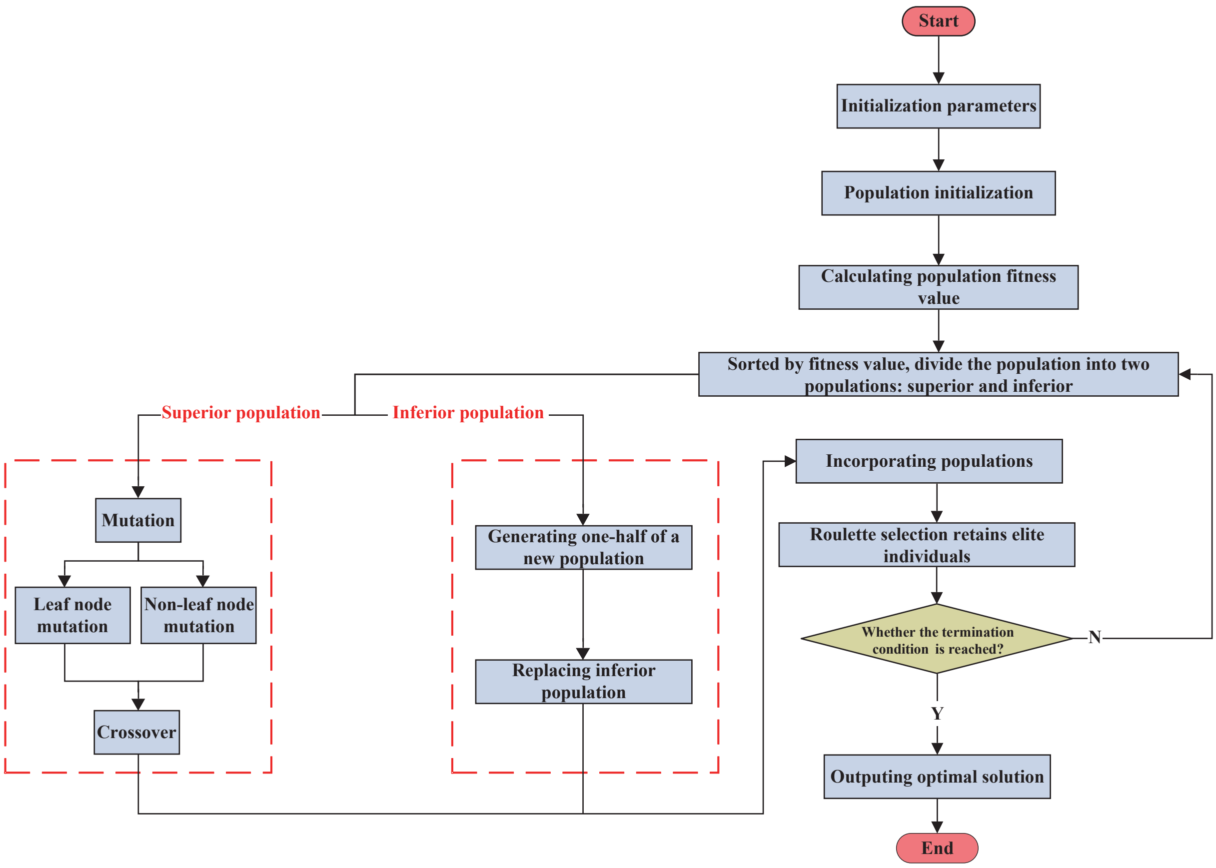


Fig. 7 IGPA flowchart.

flexible, 60% flexible, and 30% flexible. The higher the flexibility coefficient, the more operations the workshop machine can handle. The workshop machines are categorized into three sizes: $M=\{5,10,15\}$, indicating that the number of machines in workshops of different sizes are 5, 10, and 15, respectively. The number of workpieces is divided into five categories $J=\{10,20,50,100,200\}$, and to make the research closer to a real production scenario and balance the machine workload, the combination of the number of workpieces and machines is limited, and the combination mode is shown in Table 5[46]. The case follows the naming rule "the number of workpieces-number of machines-flexibility coefficient", e.g., J20M10F3 indicates 20 workpieces, 10 total machines, and 30% flexibility degree coefficient. The processing time of each workpiece in the flexible workshop varies depending on the selected machine. In this paper, the range of the operations number for each workpiece is set as [1, 2m] and the corresponding operations number is randomly distributed. Additionally, the range of operation time is set as [m/2, 2m] and each operation's processing time is randomly distributed. According to references [47] and [48], the confirmation of workpiece weights usually depends on the actual production situation of the workshop and the historical processing data of the workpieces, and a specific proportion is more in line with the actual processing situation of the workpieces. To differentiate the importance of workpieces, different weights are assigned based on the 4:2:1 principle, representing important workpieces, relatively important workpieces, and generally important workpieces. They account for 20%, 60% and 20% of the total workpieces, respectively. Different priorities result in different processing sequences, which leads to different maximum completion times for each workpiece[47].

Table 5 Workshop size information.

Scale	Number of workpieces	Number of machines
Small-scale	10	5
	20	5
	50	5
Medium-scale	20	10
	50	10
	100	10
Large-scale	50	15
	100	15
	200	15

The due date of the workpiece is determined using the total work content (TWK) rule and the due date for each workpiece is defined as follows.

$$d_i = r_i + \alpha \times \sum_{j=1}^{N_i} p_{i,j} \tag{9}$$

In Eq. (9), d_i is the due date of the i-th workpiece; r_i is the start processing time of the workpiece. If the number of workpieces does not exceed 50, then $r_i \in [0,20]$. Otherwise, $r_i=[0,40]$; α is the due-date-tightness level, and the larger the value of α, the looser the due date. $P_{i,j}$ is the processing time of operation $O_{i,j}$.

Three different due-date-tightness levels "2, 4, 8" are set to represent extremely urgent parts, urgent parts and less urgent parts, and the mean total processing time of each operation is defined as：

$$\bar{p}_{i,j} = \frac{\sum_{k=1}^{n(F(O_{i,j}))} p_{i,j,k}}{n(F(O_{i,j}))} \tag{10}$$

In Eq. (10), $P_{i,j,k}$ is the processing time of operation $O_{i,j}$ on machine M_k. $n(F(O_{i,j}))$ is the number of machines that can process operation $O_{i,j}$.

DRs as a specific representation of the workshop's characteristics. It is important to note that the same DRs should yield the same results for workshops of different sizes. In this paper, the IGPA is employed to obtain CDRs. The training set consists of small-scale workshops from Table 5. The generated CDRs are then applied to solve scheduling problems in medium-scale and large-scale workshops, in order to validate their effectiveness in FJSP scenarios.

4.2 Parameter setting

The main parameters of IGPA are the initial maximum depth of individuals (I_d), the maximum depth of individuals after crossover mutation (T_d), the crossover probability (P_c), the mutation probability (P_m), and the number of elitist preservation (E_n). In this paper, the parameters of IGPA will be set by the orthogonal test method, and four different levels of values are set for each parameter which is shown in Table 6. Selecting $L_{16}(4^5)$ orthogonal table, setting the maximum number of iterations as 100, taking the J50M10F6 as an example, each group running 20 times, and the average value of MCT is taken as the test results (which is shown in Table 7, and the unit of MCT is hour). Simultaneously, the mean value of each parameter at

Table 6 Parameter factor level.

Parameter	Parameter level			
	1	2	3	4
I_d	5	6	7	8
T_d	15	16	17	18
P_c	0.3	0.4	0.5	0.6
P_m	0.04	0.05	0.06	0.07
E_n	5	10	15	20

Table 7 Orthogonal test results.

Test number	Parameter level					Test result average(h)
	I_d	T_d	P_c	P_m	E_n	
1	5	15	0.3	0.04	5	448.16
2	5	16	0.4	0.05	10	450.34
3	5	17	0.5	0.06	15	447.51
4	5	18	0.6	0.07	20	452.22
5	6	15	0.4	0.06	20	452.39
6	6	16	0.3	0.07	15	450.16
7	6	17	0.6	0.04	10	450.3
8	6	18	0.5	0.05	5	452.39
9	7	15	0.5	0.07	10	446.5
10	7	16	0.6	0.06	5	447.56
11	7	17	0.3	0.05	20	445.27
12	7	18	0.4	0.04	15	448.15
13	8	15	0.6	0.05	15	446.56
14	8	16	0.5	0.04	20	451.1
15	8	17	0.4	0.07	5	448.86
16	8	18	0.3	0.06	10	441.91

different levels is calculated, and the change trend is shown in Fig. 8, the smaller the value of MCT is, the stronger the optimization performance of the algorithm, thus the parameters of IGPA are set as $I_d = 7$, $T_d = 17$, $P_c = 0.3$, $P_m = 0.06$, $E_n = 10$.

4.3 Analysis of result

The proposed method in this paper was implemented in Python using PyCharm 2019.3.3 x64 on a Windows 10 machine with an Intel(R) Core (TM) i5-8250U CPU @ 1.60 GHz, 16GB RAM. The comparison results between the GP algorithm and IGPA are shown in Table 8. It is evident that the IGPA is more effective than the GP algorithm. The convergence curve of the approximate solution is shown in Fig. 9, where the horizontal axis represents the number of iterations and the vertical axis represents the MCT in hours. The iteration curve of IGPA decreases steadily which verifies the effectiveness of the superior and inferior populations strategy.

According to the orthogonal tests in Section 3.2, the algorithm can get a higher performance when the number of elite reservations is 10, therefore, 10 optimal DRs combinations are selected in the form of one out of two to obtain a better DRs combination. To facilitate a comparison of the rules, five optimal rules (which are illustrated in Table 9) from the better CDRs are selected and compared its performance with five

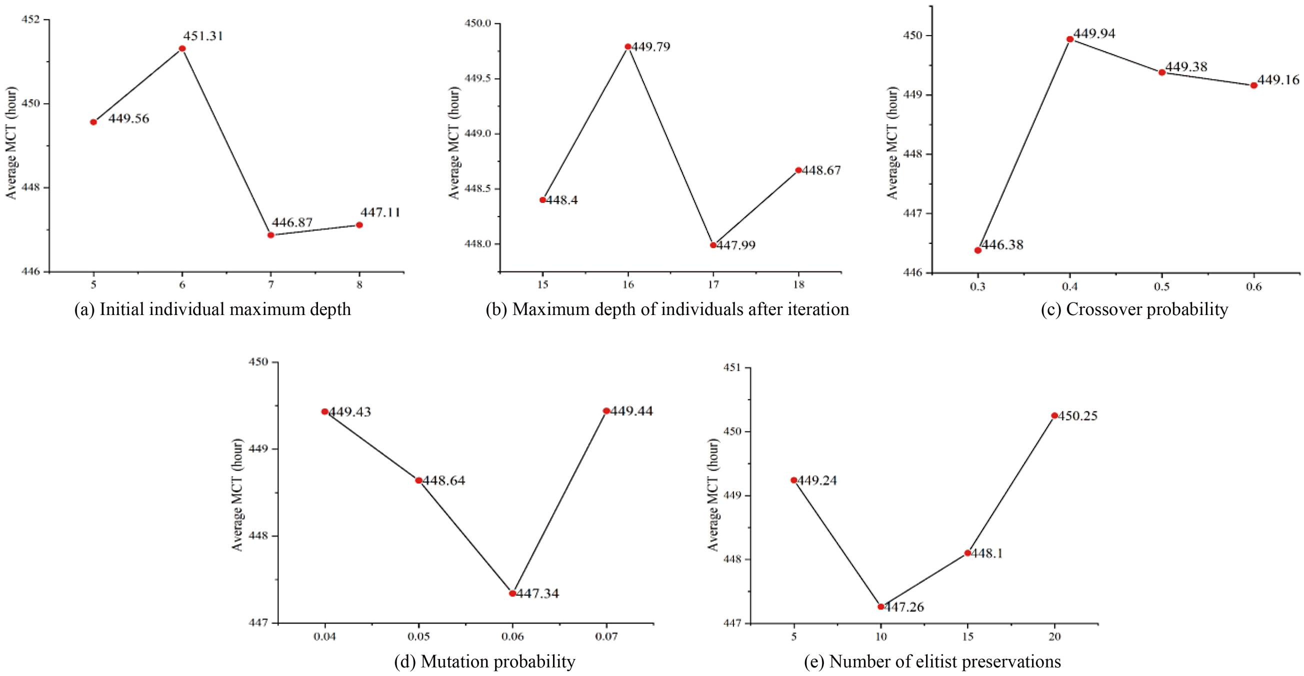
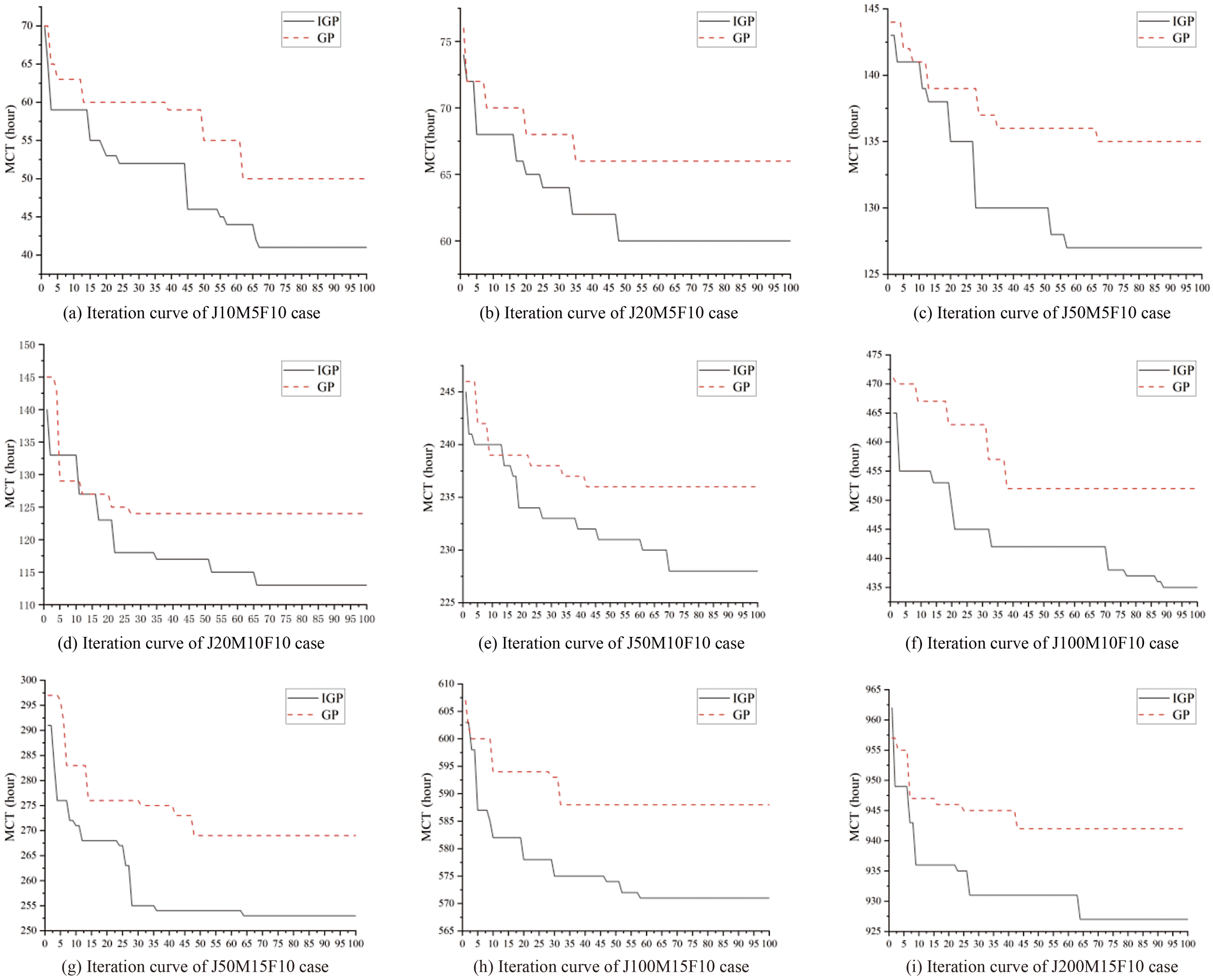


Fig. 8 Variation trend of each parameter level.

Table 8 Comparison of MCT (h).

Example scale	Example name	GP algorithm	IGPA
Small-scale	J10M5F10	50	**41**
	J20M5F10	66	**60**
	J50M5F10	135	**127**
Medium-scale	J20M10F10	124	**113**
	J50M10F10	236	**228**
	J100M10F10	452	**435**
Large-scale	J50M15F10	269	**253**
	J100M15F10	588	**571**
	J200M15F10	942	**927**



(a) Iteration curve of J10M5F10 case (b) Iteration curve of J20M5F10 case (c) Iteration curve of J50M5F10 case
(d) Iteration curve of J20M10F10 case (e) Iteration curve of J50M10F10 case (f) Iteration curve of J100M10F10 case
(g) Iteration curve of J50M15F10 case (h) Iteration curve of J100M15F10 case (i) Iteration curve of J200M15F10 case

Fig. 9 Example iteration graph.

existing SDRs (SPT, FIFO, EDD, LPT, and ST) in simulation experiments.

There are 27 different combinations of cases based on the size of the workshop and the degree of flexibility coefficient. To avoid the randomness of the experimental results, 30 groups of cases are regenerated and which have been solved by the five different CDRs and SDRs, and 30 groups of corresponding results are averaged which is presented in Table 10. It also can be seen that the CDRs outperform the SDRs.

In order to validate the effectiveness of CDR, the

Table 9 Five different CDRs.

CDRs	Rule expression
Rule_1	ATPT+2RD-CT+PT
Rule_2	min{DD/ATPT,2W*RPT}
Rule_3	CT*max{NPT,NOR}+3ATPT
Rule_4	RD*(3NOP-ATPT)
Rule_5	2CT-NPT+3RPT

Table 10 Comparison results between SDRs and CDRs.

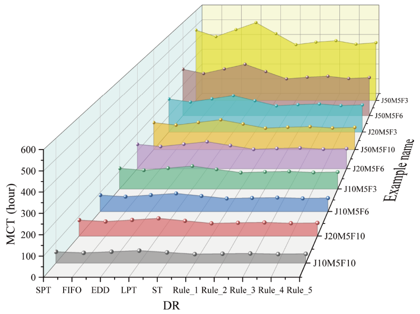
Scale	Example name	SPT	FIFO	EDD	LPT	ST	Rule_1	Rule_2	Rule_3	Rule_4	Rule_5
Small-scale	J10M5F3	115.3	104.7	116.2	127.6	109.6	**91.6**	**95.7**	**98.1**	**92.4**	**94.9**
	J10M5F6	85.7	77.8	86.3	94.8	81.4	**68.1**	**71.1**	**72.9**	**68.7**	**70.5**
	J10M5F10	53.6	48.6	54.0	59.3	50.9	**42.5**	**44.4**	**45.6**	**42.9**	**44.1**
	J20M5F3	204.0	185.3	205.5	225.8	193.9	**162.1**	**169.3**	**173.6**	**163.5**	**167.9**
	J20M5F6	139.2	126.4	140.3	154.1	132.3	**110.6**	**115.5**	**118.5**	**111.6**	**114.6**
	J20M5F10	80.3	72.9	80.9	88.9	76.3	**63.8**	**66.7**	**68.4**	**64.4**	**66.1**
	J50M5F3	470.1	426.9	473.6	520.3	446.8	**373.5**	**390.1**	**400.0**	**376.7**	**386.9**
	J50M5F6	299.9	272.3	302.1	331.9	285.0	**238.3**	**248.9**	**255.2**	**240.3**	**246.8**
	J50M5F10	160.7	145.9	161.9	177.8	152.7	**127.6**	**133.3**	**136.7**	**128.7**	**132.2**
Medium-scale	J20M10F3	378.8	344.5	380.8	420.4	361.0	**301.6**	**312.8**	**320.7**	**304.2**	**312.1**
	J20M10F6	257.9	234.6	259.3	286.2	245.8	**205.3**	**213.0**	**218.4**	**207.1**	**212.5**
	J20M10F10	149.2	135.7	149.9	165.5	142.2	**118.7**	**123.2**	**126.3**	**119.8**	**122.9**
	J50M10F3	872.9	793.9	877.4	968.7	831.9	**694.9**	**720.9**	**739.1**	**701.0**	**719.2**
	J50M10F6	556.9	506.5	559.7	618.0	530.7	**443.3**	**459.9**	**471.5**	**447.2**	**458.8**
	J50M10F10	298.3	271.3	299.9	331.1	284.3	**237.5**	**246.4**	**252.6**	**239.6**	**245.8**
	J100M10F3	1696.4	1542.8	1705.1	1882.6	1616.7	**1350.4**	**1400.9**	**1436.3**	**1362.4**	**1397.7**
	J100M10F6	1054.1	958.6	1059.5	1169.8	1004.5	**839.1**	**870.5**	**892.4**	**846.5**	**868.5**
	J100M10F10	546.9	497.4	549.7	607.0	521.2	**435.4**	**451.7**	**463.1**	**439.2**	**450.6**
Large-scale	J50M15F3	973.8	886.2	978.6	1077.9	927.9	**775.0**	**803.6**	**824.9**	**783.2**	**802.0**
	J50M15F6	621.2	565.4	624.3	687.6	591.9	**494.4**	**512.7**	**526.2**	**499.7**	**511.6**
	J50M15F10	332.8	302.9	334.4	368.4	317.1	**264.9**	**274.7**	**281.9**	**267.7**	**274.1**
	J100M15F3	1892.4	1722.3	1901.8	2094.7	1803.2	**1506.1**	**1561.8**	**1603.0**	**1522.1**	**1558.6**
	J100M15F6	1175.9	1070.2	1181.7	1301.6	1120.5	**935.8**	**970.4**	**996.1**	**945.8**	**968.5**
	J100M15F10	610.1	555.3	613.2	675.3	581.4	**485.6**	**503.5**	**516.8**	**490.7**	**502.5**
	J200M15F3	3729.7	3394.4	3748.2	4128.4	3554.0	**2968.4**	**3078.1**	**3159.4**	**2999.9**	**3071.8**
	J200M15F6	1938.5	1764.3	1948.2	2145.8	1847.2	**1542.8**	**1599.9**	**1642.1**	**1559.2**	**1596.6**
	J200M15F10	1164.8	1060.1	1170.6	1289.3	1109.9	**927.0**	**961.3**	**986.7**	**936.8**	**959.3**

data from Table 10 has been processed and analyzed from four different perspectives.

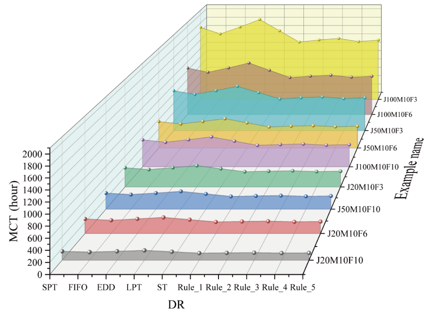
(1) Generality of CDR

To investigate the effectiveness of the same DR in workshops of different sizes, the lower-size workshop is used as the training set to obtain a better CDR which is applied to solve the medium and higher-size workshops and compared with the common SDR. Figure 10 presents the performance of each DR in three different size workshops.

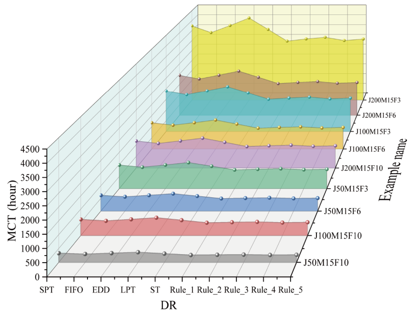
To further compare and analyze the performance difference of each DR at different workshop sizes, the variance analysis method based on change ratio is introduced to compare whether there is an overlap between the sample means of SDR and CDR at confidence intervals above 0.95. The results are shown in Fig. 11, where there is some overlap between Rule_3 and FIFO, but the optimization results are better in comparison. Rule1, Rule_2, Rule_4, Rule_5, and the other rules do not overlap and the MCT is lower than



(a) DR performance at small-scale



(b) DR performance at medium-scale



(c) DR performance at large-scale

Fig. 10 DRs performance at different scales.

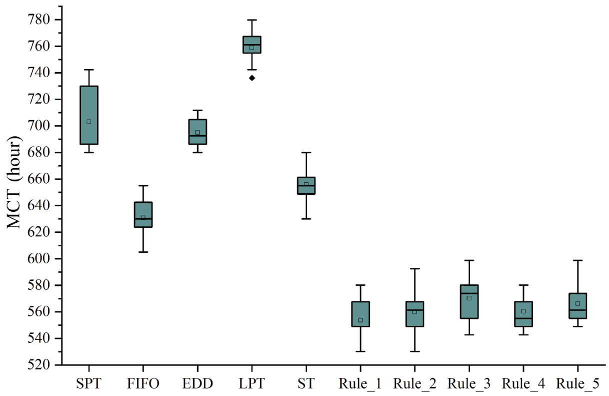


Fig. 11 Upper and lower bounds of the sample mean with the 95% confidence interval.

that of the SDR.

(2) Stability of CDR

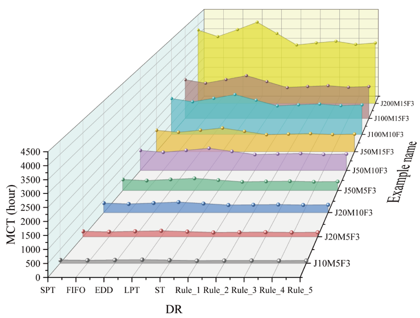
Figure 12 illustrates the performance of each DR at three different degrees of workshop flexibility coefficients (100%, 60%, and 30%). It reveals that as the workshop flexibility coefficient decreases, the objective value of each DR gradually increases. Among the various SDRs, CDRs perform better at different degrees of workshop flexibility, indicating their strong ability to resist external disturbances.

(3) Influence of delivery date margin coefficient on CDR

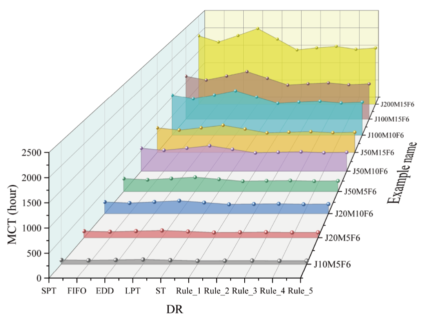
When generating extension cases, three different delivery date margin coefficients are set. The coefficients $\alpha=2$, $\alpha=4$, $\alpha=8$ represent extremely urgent workpieces, urgent workpieces, and less urgent workpieces respectively, the optimization performance is presented in Fig. 13 and the MCT for each rule is shown in Table 11. It can be seen that the delivery date margin coefficients have a minor impact on the MCT. The reason is that the delivery date margin coefficient mainly affects the mean tardiness weight of the workpiece. In addition, the FIFO rule demonstrates the best performance in optimizing the minimized MCT. The FIFO rule employs the RD to schedule workpieces to be processed, and the workpieces with an earlier RD date have a higher probability of being selected for processing compared to the workpieces with a later RD date, which ultimately reduces the total processing time.

(4) Performance of CDR on mean weighted tardiness of workpieces

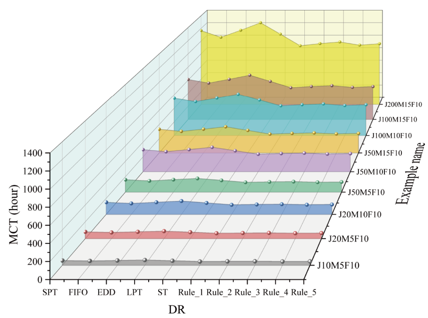
As shown in Fig. 14, the optimization performance of different DRs on the mean weighted tardiness of workpieces has been compared. With the due date margin coefficient increases, the performance of the DR gradually improves. The main reason is that as the



(a) DR performance under 30% flexible FJSP



(b) DR performance under 60% flexible FJSP



(c) DR performance under 100% flexible FJSP

Fig. 12 Performance of DRs with different flexibility coefficients.

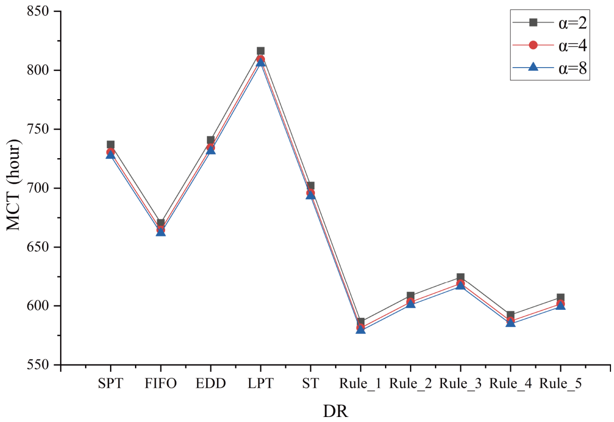


Fig. 13 Performance of DRs with different delivery date margin coefficients.

due date margin coefficient increases, the delivery date of the workpiece will also increase. The FIFO performs the worst due to the fact that it only considers the weight of the workpiece based on its arrival sequence, without considering the processing time and delivery date of the workpiece.

5 DRL-Based Analysis of the CDRs Performance

In order to further verify the superiority of the CDRs compared with the SDR in solving FJSP, the CDRs generated in Section 3 and the SDRs are taken as the optional action set of DRL in solving FJSP. Making statistics on the action selection in the DRL calculation process, i.e., the number of times CDR and SDR are selected. If CDR is selected more times than SDR, CDR has better performance. Otherwise, SDR has better performance.

5.1 State characteristics definition

Flexible job shops are complex and diverse, with a production process that involves multiple parameters. Choosing appropriate state characteristics can effectively improve the perception of the scheduling system. Two types of state characteristics are defined, one is the equipment and describes its operational status, and the other is the workpiece and describes its processing. The details are as follows.

(1) Average machine load

$$U_{\text{ave}}(t) = \frac{\sum_{k=1}^{m} U_k(t)}{m} \tag{11}$$

In Eq. (11), m is the number of processing machines and $U_k(t)$ denotes the utilization rate of machine k at time t.

$$U_k(t) = \frac{\sum_{i=1}^{n} p_{ik}}{MCT_k(t)} \tag{12}$$

In Eq. (12), P_{ik} denotes the time required for the workpiece J_i to be processed on the machine k and $MCT_k(t)$ denotes the cumulative processing time of the machine k at moment t.

(2) The ratio of the remaining processing operations at the current decision-making time to the total number of operations.

$$OR = \frac{\sum_{i=1}^{n} u_i}{m \times n} \tag{13}$$

In Eq. (13), u_i indicates the number of operations remaining for the workpiece i.

(3) The ratio of the remaining processing time of the workpiece at the current decision-making time to the total processing time.

$$TR = \frac{\sum_{i=1}^{n} \sum_{j=1}^{u} P_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} P_{ij}} \tag{14}$$

(4) The ratio of the minimum delay time to the maximum delay time at the current decision-making time.

$$DT = \frac{\min(t - DD_{ij})}{\max(t - DD_{ij})} \tag{15}$$

(5) The operation delay ratio is the ratio of the average processing time and the average relaxation time of the remaining workpieces at the current scheduling time.

$$Tard = \frac{EMPT}{EMST + \lambda} \tag{16}$$

Table 11 MCT of each rule in different delivery date margin coefficients.

α	SPT	FIFO	EDD	LPT	ST	Rule_1	Rule_2	Rule_3	Rule_4	Rule_5
2	737	670	740	816	702	586	608	624	592	607
4	730	664	734	809	695	581	603	618	587	601
8	727	661	731	805	693	578	600	616	584	599

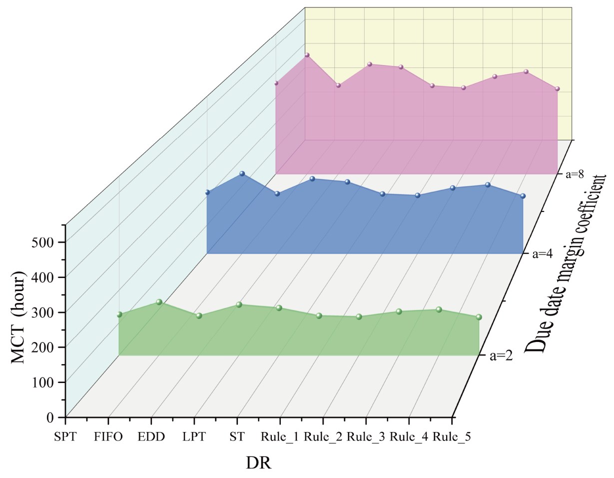


Fig. 14 Optimal performance of DRs for mean tardiness of weight.

$$EMPT = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{u} p_{ij} \tag{17}$$

$$EMST = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{u} p_{ij} + PST - \mathrm{DD}_i\right) \tag{18}$$

where PST denotes the current decision moment, u denotes the number of remaining operations, and λ is a positive number that guarantees the significance of the characteristic parameters.

5.2 Action set selection

The action set contains all the possible behaviors generated by the agent at the decision point, it is the sum of all the optional executable actions of the agent, and the agent guides the production scheduling of the workshop by selecting the actions. In this paper, five SDRs and the CDRs obtained in Section 3 are selected to construct the action set. These are presented in Tables 12 and 13, respectively.

The scheduling objective of this paper is to minimize the MCT, therefore, in order to facilitate the calculation of the reward function, the reward function is designed as follows:

$$r_{sq} = -C_{iks} \tag{19}$$

where r_{sq} is the reward obtained after assigning workpiece J_i to equipment M_k for processing at each scheduling decision point in stage s, and C_{iks} is the completion time of workpiece J_i in stage s. The rewards designed in this paper are negative, and the purpose of the agent in the execution process is to avoid getting negative rewards, so the maximum value of the total reward is 0.

Table 12 SDR action set.

Action	Rule name	Explanation
action1	SPT Rule	Prioritize operations with short processing times
action2	EDD Rule	Priority processing of operations with an early due date
action3	FIFO Rule	Prioritize operations that arrive at the equipment earlier
action4	OSL Rule	Prioritize operations with low slack time
action5	LPT Rule	Prioritize operations with long processing times

Table 13 CDR action set.

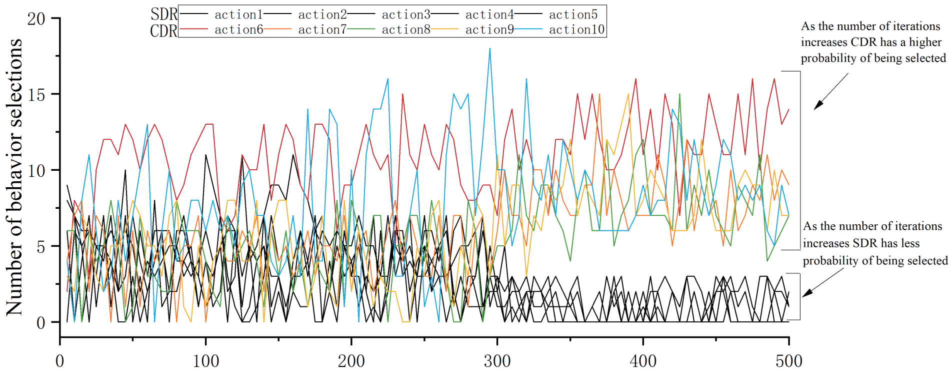
Action	Rule name	Rule expression
action6	Rule_1	ATPT+2RD-CT+PT
action7	Rule_2	min{DD/ATPT,2W*RPT}
action8	Rule_3	CT*max{NPT,NOR}+3ATPT
action9	Rule_4	RD*(3NOP-ATPT)
action10	Rule_5	2CT-NPT+3RPT

5.3 Analysis of result

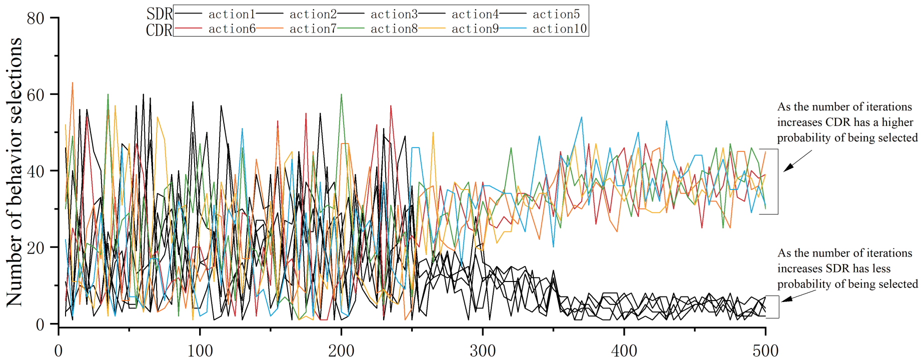
To further verify the superior performance of the CDRs, the selection of actions in the calculation process of the arithmetic case is counted. The probability of being selected by SDRs and CDRs is compared to demonstrate the superior performance of the CDRs. By analyzing the iteration process of the cases J10M5F3, J20M10F3, and J50M15F3, the frequency of selecting ten rules by DRL is recorded, as shown in Fig. 15. As the number of training increases, there is a gradual decrease in the selection of actions action 1 – action 5, which represent SDRs. Conversely, there is a gradual increase in the selection of actions action 6 – action 10, which represent the CDRs. These figures demonstrate the superior performance of the CDRs in DQL by acting as action sets. Additionally, it highlights the effectiveness of using CDRs simultaneously to generate suitable scheduling schemes.

6 Conclusion and Future Work

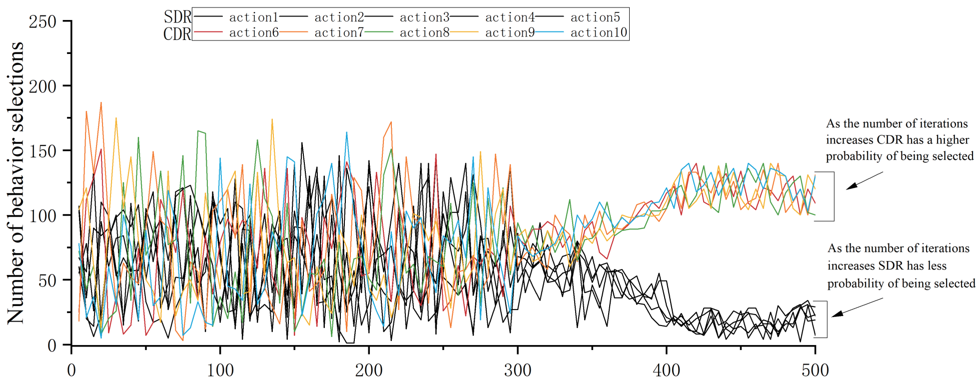
In this paper, an IGPA has been proposed to mine the CDRs for FJSP. To prevent the algorithm from falling into local optimal solutions, the superior and inferior population separation strategy is designed. Additionally, a normalization method is introduced to normalize the terminator set to reduce unintended solutions during the optimization process. The leaf node mutation and non-leaf node mutation strategy are



(a) Schematic diagram of selected times in J10M5F3



(b) Schematic diagram of selected times in J20M10F3



(c) Schematic diagram of selected times in J50M15F3

Fig. 15 Schematic diagram of the number of selected DRs under different scales.

designed to enhance the algorithm's local search ability. The algorithm parameters are determined by applying orthogonal tests, and 30 test cases are generated for comparing the optimized CDRs with the SDRs. The comparison results effectively demonstrate the advantages of CDRs over SDRs and also highlight the effectiveness of the IGPA in CDRs evolving. Finally, in order to further verify the superiority of the CDRs, the DRL is employed to solve the FJSP by incorporating the CDRs as an action set, and the statistical analysis of the selection times is illustrated. For future work, the priority of the machine can be introduced into the DRs to reduce the differences existing between the shop scheduling model and the manufacturing system. Meanwhile, other optimization strategies can be introduced to mine the DRs to further improve the solution efficiency of the algorithm.

Acknowledgment

This research was supported by the National Natural

Science Foundation of China (Nos. 51805152 and 52075401), the Green Industry Technology Leading Program of Hubei University of Technology (No. XJ2021005001), and the Scientific Research Foundation for High-level Talents of Hubei University of Technology (No. GCRC2020009), the Natural Science Foundation of Hubei Province (No. 2022CFB445).

References

[1] R. Li, W. Gong, L. Wang, C. Lu, and X. Zhuang, Surprisingly popular-based adaptive memetic algorithm for energy-efficient distributed flexible job shop scheduling, *IEEE Trans. Cybern.*, vol. 53, no. 12, pp. 8013–8023, 2023.

[2] E. Jiang, L. Wang, and J. Wang, Decomposition-based multi-objective optimization for energy-aware distributed hybrid flow shop scheduling with multiprocessor tasks, *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 646–663, 2021.

[3] Z. Chen, L. Zhang, X. Wang, and P. Gu, Optimal design of flexible job shop scheduling under resource preemption based on deep reinforcement learning, *Complex Syst. Model. Simul.*, vol. 2, no. 2, pp. 174–185, 2022.

[4] H. Zhang, J. Xie, J. Ge, J. Shi, and Z. Zhang, Hybrid particle swarm optimization algorithm based on entropy theory for solving DAR scheduling problem, *Tsinghua Sci. Technol.*, vol. 24, no. 3, pp. 282–290, 2019.

[5] X. Li, X. Guo, H. Tang, R. Wu, L. Wang, S. Pang, Z. Liu, W. Xu, and X. Li, Survey of integrated flexible job shop scheduling problems, *Comput. Ind. Eng.*, vol. 174, p. 108786, 2022.

[6] L. Meng, W. Cheng, B. Zhang, W. Zou, W. Fang, and P. Duan, An improved genetic algorithm for solving the multi-AGV flexible job shop scheduling problem, *Sensors*, vol. 23, no. 8, p. 3815, 2023.

[7] X. Long, J. Zhang, K. Zhou, and T. Jin, Dynamic self-learning artificial bee colony optimization algorithm for flexible job-shop scheduling problem with job insertion, *Processes*, vol. 10, no. 3, p. 571, 2022.

[8] J. Li, X. Gu, Y. Zhang, and X. Zhou, Distributed flexible job-shop scheduling problem based on hybrid chemical reaction optimization algorithm, *Complex Syst. Model. Simul.*, vol. 2, no. 2, pp. 156–173, 2022.

[9] J. Duan and J. Wang, Robust scheduling for flexible machining job shop subject to machine breakdowns and new job arrivals considering system reusability and task recurrence, *Expert Syst. Appl.*, vol. 203, p. 117489, 2022.

[10] D. Lei, M. Li, and L. Wang, A two-phase meta-heuristic for multiobjective flexible job shop scheduling problem with total energy consumption threshold, *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1097–1109, 2019.

[11] Z. Zhuang, Y. Li, Y. Sun, W. Qin, and Z.-H. Sun, Network-based dynamic dispatching rule generation mechanism for real-time production scheduling problems with dynamic job arrivals, *Robot. Comput. Integr. Manuf.*, vol. 73, p. 102261, 2022.

[12] H. Zhang and U. Roy, A semantics-based dispatching rule selection approach for job shop scheduling, *J. Intell. Manuf.*, vol. 30, no. 7, pp. 2759–2779, 2019.

[13] C.-H. Lee A dispatching rule and a random iterated greedy metaheuristic for identical parallel machine scheduling to minimize total tardiness, *Int. J. Prod. Res.*, vol. 56, no. 6, pp. 2292–2308, 2018.

[14] J. Branke, S. Nguyen, C. W. Pickardt, and M. Zhang, Automated design of production scheduling heuristics: A review, *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 110–124, 2016.

[15] M. Thenarasu, K. Rameshkumar, J. Rousseau, and S. P. Anbuudayasankar, Development and analysis of priority decision rules using MCDM approach for a flexible job shop scheduling: A simulation study, *Simul. Model. Pract. Theory*, vol. 114, p. 102416, 2022.

[16] C. Ferreira, G. Figueira, and P. Amorim, Effective and interpretable dispatching rules for dynamic job shops via guided empirical learning, *Omega*, vol. 111, p. 102643, 2022.

[17] D. Wu, G. Zeng, D. He, Z. Qian, and Q. Zhang, Task coordination organization model and the task allocation algorithm for resource contention of the syncretic system, *Tsinghua Sci. Technol.*, vol. 21, no. 4, pp. 459–470, 2016.

[18] S. Jun, S. Lee, and H. Chun, Learning dispatching rules using random forest in flexible job shop scheduling problems, *Int. J. Prod. Res.*, vol. 57, no. 10, pp. 3290–3310, 2019.

[19] B. Chen and T. I. Matis, A flexible dispatching rule for minimizing tardiness in job shop scheduling, *Int. J. Prod. Econ.*, vol. 141, no. 1, pp. 360–365, 2013.

[20] B. Mihoubi, B. Bouzouia, and M. Gaham, Reactive scheduling approach for solving a realistic flexible job shop scheduling problem, *Int. J. Prod. Res.*, vol. 59, no. 19, pp. 5790–5808, 2021.

[21] J. Heger and T. Voss, Reducing mean tardiness in a flexible job shop containing AGVs with optimized combinations of sequencing and routing rules, *Procedia CIRP*, vol. 81, pp. 1136–1141, 2019.

[22] A. Teymourifar and G. Ozturk, New dispatching rules and due date assignment models for dynamic job shop scheduling problems, *Int. J. Manuf. Res.*, vol. 13, no. 4, p. 302, 2018.

[23] M. Đurasević and D. Jakobović, Selection of dispatching rules evolved by genetic programming in dynamic unrelated machines scheduling based on problem characteristics, *J. Comput. Sci.*, vol. 61, pp. 101649, 2022.

[24] L. Wang, Z. Pan, and J. Wang, A review of reinforcement learning based intelligent optimization for manufacturing scheduling, *Complex Syst. Model. Simul.*, vol. 1, no. 4, pp. 257–270, 2021.

[25] K. Chang, S. H. Park, and J.-G. Baek, AGV dispatching algorithm based on deep Q-network in CNC machines environment, *Int. J. Comput. Integr. Manuf.*, vol. 35, no. 6, pp. 662–677, 2022.

[26] W. Song, X. Chen, Q. Li, and Z. Cao, Flexible job-shop scheduling via graph neural network and deep reinforcement learning, *IEEE Trans. Ind. Inform.*, vol. 19, no. 2, pp. 1600–1610, 2023.

[27] C.-L. Liu, C.-J. Tseng, T.-H. Huang, and J.-W. Wang, Dynamic parallel machine scheduling with deep Q-

network, *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 53, pp. 6792–6804, 2023.

[28] Y. Gui, D. Tang, H. Zhu, Y. Zhang, and Z. Zhang, Dynamic scheduling for flexible job shop using a deep reinforcement learning approach, *Comput. Ind. Eng.*, vol. 180, p. 109255, 2023.

[29] B. Min and C. O. Kim, State-dependent parameter tuning of the apparent tardiness cost dispatching rule using deep reinforcement learning, *IEEE Access*, vol. 10, pp. 20187–20198, 2022.

[30] M. M. Nasiri, R. Yazdanparast, and F. Jolai, A simulation optimisation approach for real-time scheduling in an open shop environment using a composite dispatching rule, *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 12, pp. 1239–1252, 2017.

[31] A. Yadav and S. C. Jayswal, Modelling of flexible manufacturing system: a review, *Int. J. Prod. Res.*, vol. 56, no. 7, pp. 2464–2487, 2018.

[32] R. Li, W. Gong, and C. Lu, Self-adaptive multi-objective evolutionary algorithm for flexible job shop scheduling with fuzzy processing time, *Comput. Ind. Eng.*, vol. 168, p. 108099, 2022.

[33] E.-D. Jiang and L. Wang, Multi-objective optimization based on decomposition for flexible job shop scheduling under time-of-use electricity prices, *Knowl. Based Syst.*, vol. 204, p. 106177, 2020.

[34] M. Đurasević and D. Jakobović, Evolving dispatching rules for optimising many-objective criteria in the unrelated machines environment, *Genet. Program. Evolvable Mach.*, vol. 19, no. 1, pp. 9–51, 2018.

[35] S. Shady, T. Kaihara, N. Fujii, and D. Kokuryo, Feature selection approach for evolving reactive scheduling policies for dynamic job shop scheduling problem using gene expression programming, *Int. J. Prod. Res.*, vol. 61, no. 15, pp. 5029–5052, 2023.

[36] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, Automatic programming via iterated local search for dynamic job shop scheduling, *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 1–14, 2015.

[37] B. Xu, Y. Mei, Y. Wang, Z. Ji, and M. Zhang, Genetic programming with delayed routing for multiobjective dynamic flexible job shop scheduling, *Evol. Comput.*, vol. 29, no. 1, pp. 75–105, 2021.

[38] L. Zhang, Y. Hu, C. Wang, Q. Tang, and X. Li, Effective dispatching rules mining based on near-optimal schedules in intelligent job shop environment, *J. Manuf. Syst.*, vol. 63, pp. 424–438, 2022.

[39] S. Jun and S. Lee, Learning dispatching rules for single machine scheduling with dynamic arrivals based on decision trees and feature construction, *Int. J. Prod. Res.*, vol. 59, no. 9, pp. 2838–2856, 2021.

[40] S. Wang, H. Su, G. Wan, and L. Zhong, Surgery scheduling in the presence of operating room eligibility and dedicated surgeon: An adaptive composite dispatching method, *Int. J. Prod. Res.*, vol. 61, no. 6, pp. 1866–1881, 2023.

[41] S. Nguyen, M. Zhang, and K. C. Tan, Surrogate-assisted genetic programming with simplified models for automated design of dispatching rules, *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2951–2965, 2017.

[42] Y. Mei, S. Nguyen, B. Xue, and M. Zhang, An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming, *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 1, no. 5, pp. 339–353, 2017.

[43] J. R. Koza Genetic programming as a means for programming computers by natural selection, *Stat. Comput.*, vol. 4, no. 2, pp. 87–112, 1994.

[44] L. Zhang, N. R. Alharbe, G. Luo, Z. Yao, and Y. Li, A hybrid forecasting framework based on support vector regression with a modified genetic algorithm and a random forest for traffic flow prediction, *Tsinghua Sci. Technol.*, vol. 23, no. 4, pp. 479–492, 2018.

[45] Y. Zeiträg, J. R. Figueira, N. Horta, and R. Neves, Surrogate-assisted automatic evolving of dispatching rules for multi-objective dynamic job shop scheduling using genetic programming, *Expert Syst. Appl.*, vol. 209, p. 118194, 2022.

[46] 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, p. 113348, 2020.

[47] C.-C. Wu, S.-C. Liu, C. Zhao, S.-Z. Wang, and W.-C. Lin, A multi-machine order scheduling with learning using the genetic algorithm and particle swarm optimization, *Comput. J.*, vol. 61, no. 1, pp. 14–31, 2018.

Xixing Li received the MS and PhD degrees in mechanical engineering from Wuhan University of Technology, Wuhan, China, in 2014 and 2017, respectively. He is currently an associate Professor with School of Mechanical Engineering, Hubei University of Technology, Wuhan. He has published about 40 journal papers. His current research interests include production planning and scheduling, manufacturing informatization, and optimization modeling.

Qingqing Zhao is an undergraduate student with School of Mechanical Engineering, Hubei University of Technology, Wuhan. His research interests include shop scheduling and optimization algorithms.

Hongtao Tang received the BEng degree from Wuhan University of Technology, China in 2008, and the PhD degree from HUST, China in 2014. He is currently an associate professor at WHUT. His research interests include intelligent manufacturing and mechanical engineering.

Xing Guo is a graduate student with the School of Mechanical Engineering, Hubei University of Technology, Wuhan. His research interests include deep learning, reinforcement learning, and shop scheduling.

Mengzhen Zhuang received the BEng degree from Wuhan Textile University, China in 2021. He is currently studying at Wuhan University of Technology. His research direction is intelligent algorithm and job shop scheduling.

Yibing Li received the MS and PhD degrees from Wuhan University of Technology, Wuhan, China, in 2014 and 2017, respectively. His research interests include intelligent manufacturing, shop scheduling, artificial intelligence, optimization algorithms, and mechanical equipment fault diagnosis.

Xi Vincent Wang received the bachelor degree in mechanical engineering from Tianjin University, China in 2008, and the PhD degree from the University of Auckland, New Zealand in 2013. He is currently is an associate professor at KTH Sweden. He serves as the associate editor of *SME Journal of Manufacturing Systems*. His research interests include Cloud-based manufacturing, robotics, and digital twin.