Journal of Manufacturing Systems 83 (2025) 357–371

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

Flexible pallet automation system scheduling with limited fixture-pallets and material-pallets: A case study from an engine manufacturing enterprise

Yulu Zhou [a,b], Shichang Du [a,*], Jun Lv [c], Xiaoxiao Shen [a], Andrea Matta [b], Siyang Wang [d]

[a] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China
[b] Department of Mechanical Engineering, Politecnico di Milano, Milan, Italy
[c] Faculty of Economics and Management, East China Normal University, Shanghai, China
[d] Weichai Power Co., Ltd, Shandong, China

ARTICLE INFO

Keywords:
Pallet automation system
Fixture
Scheduling
Priority rule
Meta-heuristic algorithm
Manufacturing enterprise

ABSTRACT

Pallet automation system (PAS) is crucial for enterprises to organize and schedule limited resources, such as fixture-pallets (FPs) and material-pallets (MPs). In customized production, FPs are often insufficient and unbalanced. To address this, MPs are prepared to store workpieces to release FPs' capacity. In this way, FPs are utilized for processing, while MPs are leveraged for storage. However, existing studies mainly focus on fixtures that are fixed to machines and rarely consider FPs and MPs. To address this gap, this paper investigates the flexible pallet automation system scheduling with limited FPs and MPs (FPASFM). Firstly, a mathematical model is established to minimize the makespan. Secondly, a five-layer encoding strategy, a new decoding method, and a feasibility correction strategy are integrated to obtain feasible solutions. Thirdly, an improved meta-heuristic algorithm with rule-based initialization and critical path mutation (IMHRC) is proposed. Finally, effective initialization rule combinations are identified through experiments with 36 different rule combinations. 15 real-data case studies show that IMHRC outperforms six other algorithms. Additionally, IMHRC significantly reduces makespan by 59.66 % and 45.90 % for two real orders, while enhancing resource utilization. IMHRC demonstrates the ability to obtain superior solutions in a shorter time, with its advantages in large-scale problems, effectively meeting the practical demands of enterprises in real-world production environments.

1. Introduction

With the increasing demand for product customization and shorter product life cycles, manufacturing paradigms are shifting from high-volume and low-variety to low-volume and high-variety. In this context, flexible automation upgrades have become a critical strategy for enhancing competitiveness [1], with pallet automation systems (PASs) emerging as a key solution [2]. By employing standardized pallets and a high-speed stacker crane, PAS facilitates efficient storage, organization, and scheduling of limited flexible resources. It enables fixtures to be shared across all machines and restricts loading and unloading workpieces to setup stations (STs), thereby reducing non-processing time associated with fixture replacement and machine adjustment. A leading Chinese engine manufacturing enterprise investigated in this paper has applied various PASs in their prototype production to enhance the flexibility and adaptability of production lines, enabling itself to rapidly respond to process route changes and meet diverse customer demands [3].

In PAS, fixture-pallets (FPs) and material-pallets (MPs) are two critical resources. The varying demands of different production tasks on flexible FPs complicate scheduling due to potential shortages and imbalances in FPs' allocation. To address this, MPs are applied to store workpieces, thereby freeing up FPs' capacity, as FPs can only be used to clamp workpieces that require processing. For example, consider a workpiece clamped on the FP that has just finished processing on the machine. If the next operation requires another machine, which is currently occupied, keeping the workpiece on the FP would block the reuse of that FP. By transferring the workpiece to an MP, the FP can be released immediately and reassigned to other workpieces. This mechanism is widely implemented in industrial PAS. As shown in Fig. 1, the system is equipped with a large buffer area that stores both FPs and MPs, and a high-speed stacker crane that enables fast transportation and switching between them. FPs restrict the production sequence of different operations, and MPs limit the number of workpieces that can

* Corresponding author:.
E-mail address: lovbin@sjtu.edu.cn (S. Du).

https://doi.org/10.1016/j.jmsy.2025.09.015
Received 20 May 2025; Received in revised form 26 August 2025; Accepted 22 September 2025
Available online 26 September 2025
0278-6125/© 2025 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Fig. 1. Real PAS in a leading Chinese engine manufacturing enterprise.

Table 1
Summary of related research and this work.

Literature	Method	Pallet	fixture	MP
[27]	Meta-heuristic algorithm	√		
[28]		√		
[29]		√		
[30]			√	
[31]			√	
[32]			√	
[33]			√	
[34]			√	
[35]			√	
[36]		√	√	
[37]		√	√	
[38]	Priority rules	√	√	
[39]		√	√	
[40]		√	√	
Ours	Meta-heuristic algorithm with priority rules	√	√	√

be produced simultaneously in PAS. Therefore, the joint scheduling of FPs and MPs is crucial for improving production efficiency.

However, such a complex scheduling problem has yet to be studied. Job shop scheduling problem (JSP) typically considers only machine resources as single-resource constraints [4]. In contrast, the multi-resource constrained flexible job shop scheduling problem (MRFJSP), being extensions of JSP that better reflect real production scenarios, has attracted significant research attention [5–10]. While previous studies about MRFJSP have investigated various resources, including workers [11–15], buffers [16–19], tools [20,21], AGVs [22–24], and robots [25,26], there is limited research on FPs and MPs. Table 1 presents studies on MRFJSP related to fixture, pallet, and MP resource constraints. While Liao et al. [27], Mati et al. [28], and Diabat et al. [29] investigated limited pallet resources, other researchers considered fixture availability [30–35]. Some researchers simultaneously considered fixtures and pallets, but they ignored key constraints. Liu et al. [36] failed to consider fixtures to be used among various machines. Sim and Lee [37], Shin et al. [38], Lee et al. [39], and Yu et al. [40] neglected fixture flexibility.

Although various approaches have been proposed to address FJSP, including exact methods [41–43], priority rules [44,45], meta-heuristic algorithms [46–51], and machine learning tools [52–54], the majority of existing studies on MRFJSP primarily employ meta-heuristic algorithms and priority rules, as summarized in Table 1. Exact methods provide optimal solutions but are only feasible for small-scale problems due to their high computational cost. Machine learning approaches, while promising, require large datasets and are often difficult to interpret, limiting their practical application. Priority rules are efficient and simple, but their effectiveness depends on the combination of rules, and they lack global search capabilities. In contrast, meta-heuristic algorithms offer greater flexibility and scalability, making them well-suited for large-scale, multi-resource constrained scheduling problems.

Based on the above analysis, the research gaps can be summarized as follows:

1) Practical urgency: In the studied engine manufacturing enterprise, MPs have recently been introduced to complement FPs in PAS. However, managing multiple resources—materials, fixtures, pallets, and STs—remains challenging. Despite extensive manual scheduling efforts, machine idle rates are high, and delivery delays persist. This highlights an urgent need for an efficient algorithm to generate rapid, high-quality schedules and improve resource utilization.
2) Problem formulation gap: Most existing FJSP studies focus solely on machines, with limited attention to FPs and little to no consideration of MPs. In practice, the interdependence between FPs and MPs significantly affects scheduling. At present, no work has addressed MRFJSP considering both constraints, leaving a critical research gap.
3) Methodological limitation: Current meta-heuristic methods struggle with this problem due to insufficient encoding/decoding schemes and poor quality initial solutions. New strategies are needed to enhance feasibility and performance, better aligning with industrial scheduling needs.

Hence, this paper studied a flexible pallet automation system scheduling with limited fixture-pallets and material-pallets (FPASFM). A graphical abstract highlighting the main elements of this study is shown in Fig. 2. The effective scheduling of multiple limited flexible resources in PAS, particularly FPs and MPs, presents a significant challenge due to the need to simultaneously select appropriate resources for each operation and optimize operation sequencing. To overcome these challenges, this study offers the following contributions.

1) An improved meta-heuristic algorithm with rule-based initialization and critical path mutation (IMHRC) is developed, whose two main innovations lie in:
   a) Rule-based initialization strategy: 12 customized priority rules, forming 36 combined rule strategies, are designed for 6 initialization stages.
   b) Critical path-based mutation for local search: A new critical path identification method that includes MP allocation, loading, unloading, and processing operations is proposed. Three mutation operators for optimizing MP allocation, machine selection, and FP selection are designed.
2) A five-layer encoding scheme, a decoding method with time period insertion based on the intersection of the available time of multiple resources (TPI-IAR), and a feasibility correction strategy are applied to obtain a feasible schedule solution.
3) In the case study from real data, IMHRC demonstrates the ability to obtain superior solutions in a shorter time, with its advantages in large-scale problems. Furthermore, IMHRC reduces the makespan by 59.66 % and 45.90 % for the two real orders compared to the original scheduling solutions.

The remainder of this paper is structured as follows. Section 2 elaborates on the system description, problem definition, and mathematical modeling. Section 3 details the methodology and implementation of the proposed IMHRC algorithm. Section 4 validates the approach through three case studies. The paper concludes in Section 5

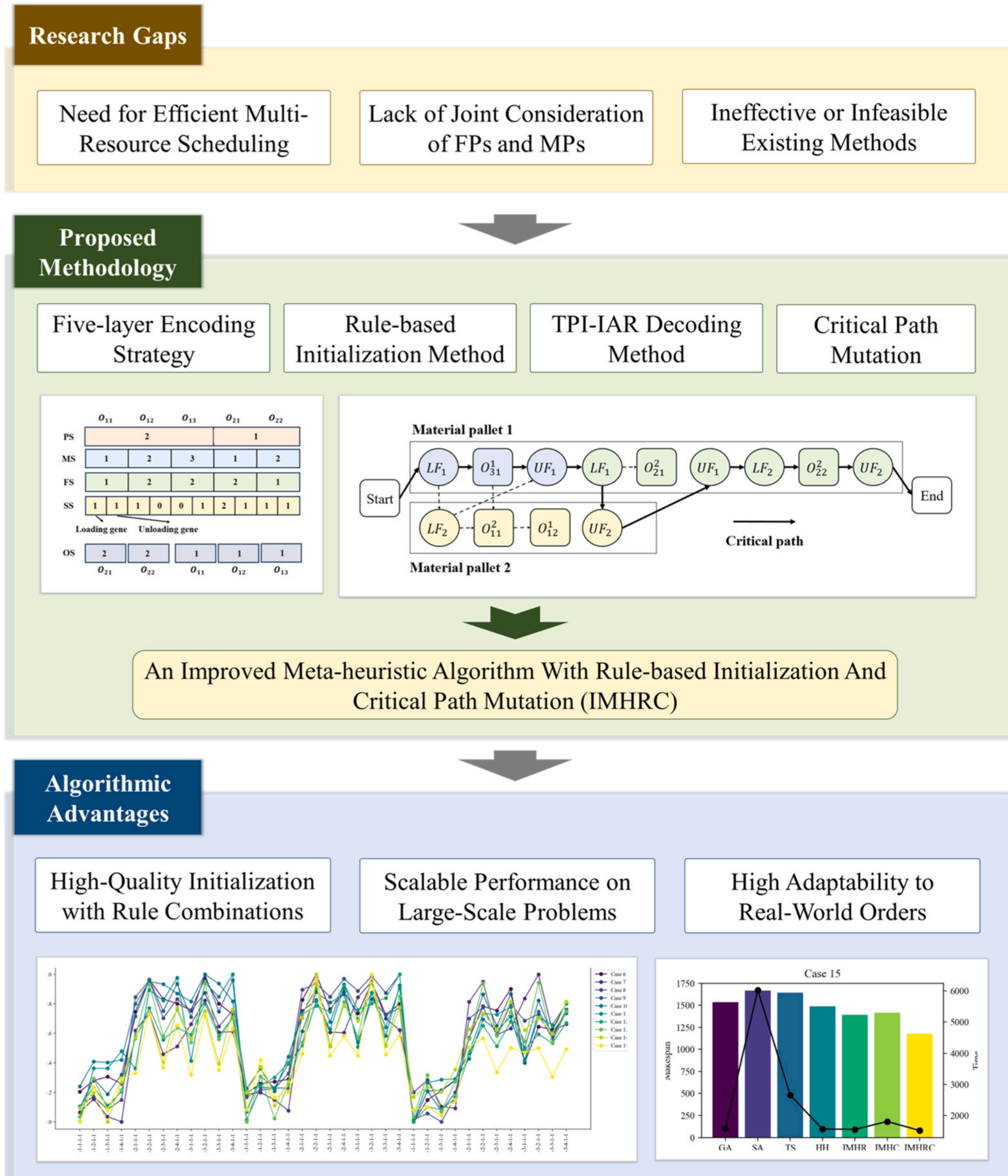

Fig. 2. The graphical abstract of this study.

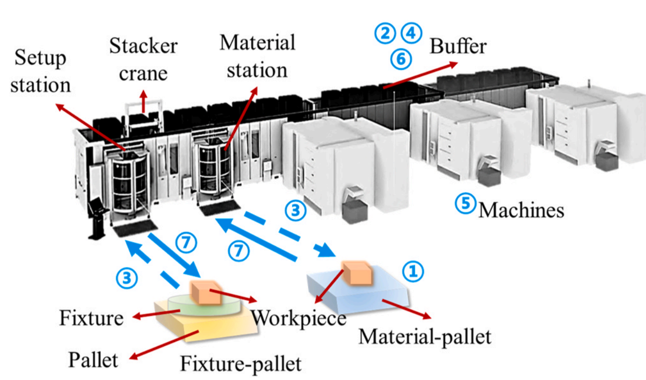

Fig. 3. The configuration of PAS.

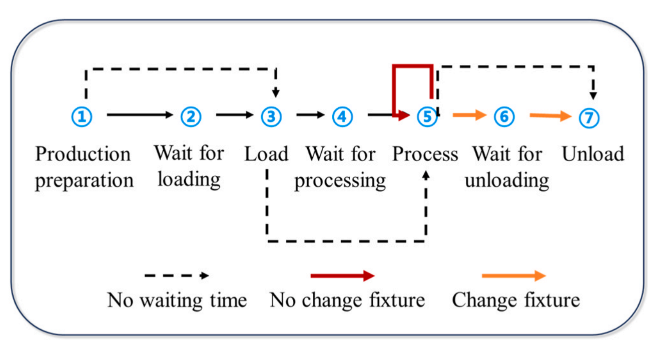

Fig. 4. Workpiece flow.

Table 2
Small case of FPASFM.

W	O	M			FPs	MPs
		M_1	M_2	M_3		
1	O_{11}	5	6	5	F_1	P_1,P_2
	O_{12}	-	5	-	F_2	
	O_{13}	5	-	2	F_1,F_2	
2	O_{21}	2	-	6	F_1,F_2	
	O_{22}	7	5	8	F_1	
ST		ST_1				
MS		MS_1				

Table 3
Loading or unloading time of each workpiece on FPs.

FP	Loading time or unloading time
F_1	5
F_2	3

with a discussion of major contributions and recommendations for further investigation.

2. The model of FPASFM

2.1. System description

The PAS integrates machines, FPs, MPs, material stations (MSs), STs, buffers, a stacker crane, and a control system (see Fig. 3). MPs hold raw and semi-finished workpieces, while FPs clamp workpieces during processing on the machines. MSs and STs facilitate loading and unloading workpieces onto MPs and FPs, respectively.

The workpiece flow under PAS is illustrated in Fig. 4, where the numbers (from ① to ⑦) correspond to those in Fig. 3. Initially, raw workpieces (RWs) are loaded onto MPs. Each fixture is paired with a pallet to form an FP. The RW may then wait in the MP for the availability of the appropriate ST, MS, and FP before being loaded. Once transferred from the MP to the FP, the workpiece may wait in the FP for an available machine for processing. After processing, it is stored in the FP, awaiting available ST, MS, and MP for unloading. The workpiece is then moved from FP to MP for the subsequent operation. The PAS outputs the finished workpiece after processing all operations and subsequently inputs a new raw workpiece. Note that if all necessary resources are available, waiting time is eliminated. Furthermore, if the same FP is required for the next operation, the workpiece can proceed directly to the next machine without loading and unloading. Otherwise, the workpiece must be unloaded from the current FP and reloaded onto the FP required for the next operation.

2.2. Problem definition

A small-scale case involving five operations, three machines, two FPs, two MPs, one ST, and one MS is utilized to clarify FPASFM, as shown in Table 2, with the loading or unloading time provided in Table 3. Each operation has various alternative machines and FPs, and all workpieces are compatible with either of the two MPs. The main constraints are illustrated as follows:

1) The processing of workpieces needs FPs and machines; The loading and unloading of workpieces need MPs, FPs, STs, and MSs.
2) All pallets are divided into FPs and MPs. An FP consists of a fixture and a pallet, while an MP only has a pallet.
3) Each operation has various available machines and FPs, and each workpiece only needs one MP.
4) Each machine, FP, MP, ST, and MS, can only be used for one operation at a time.

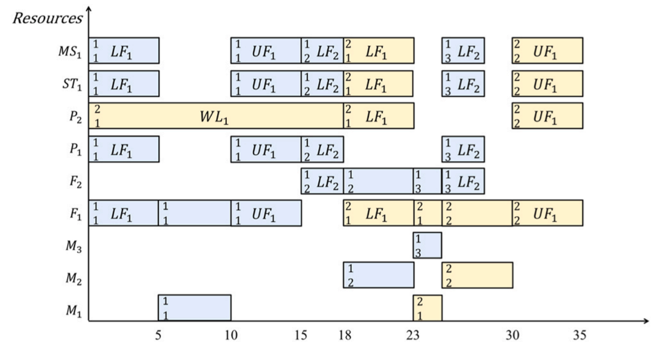

Fig. 5. Scheduling plan of the small case.

Table 4
Notations used in the model.

Parameters	Definition
I	Workpiece set indexed by i, $i = 1,2...\|I\|$
J_i	Operation set of workpiece i indexed by j,$j = 1,2...\|J_i\|$
O_{ij}	jth operation of workpiece i,$i \in I,j \in J_i$
M	Machine total set indexed by m, $m = 1,2...\|M\|$
M_{ij}	Alternative machine set of operation O_{ij},$M_{ij} \in M$
F	FP total set indexed by f, $f = 1,2...\|F\|$
F_{ij}	Alternative FP set of operation O_{ij},$F_{ij} \in F$
P	MP total set indexed by p, $p = 1,2...\|P\|$
N	ST used for setting FP indexed by n, $n = 1,2...\|N\|$
T_{ijm}	Processing time of operation O_{ij}on machine m,$i \in I,j \in J_i,m \in M_{ij}$
E_f	Loading time or unloading time of any workpiece on FP f,$f \in F$
L	A large number, $L > 0$
Time variables	
$C_{\max}$	Maximum completion time (makespan), $C_{\max} > 0$
S_{ij}	Start processing time of operation O_{ij}
C_{ij}	End processing time of operation O_{ij}
SLF_{ij}	Start loading time of operation O_{ij}
CLF_{ij}	End loading time of operation O_{ij}
SUF_{ij}	Start unloading time of operation O_{ij}
CUF_{ij}	End unloading time of operation O_{ij}
SWL_{ij}	Start waiting loading time of operation O_{ij}
CWL_{ij}	End waiting loading time of operation O_{ij}
0−1 decision variables	
D_{ip}	If MP p is used by workpiece$i$,$D_{ip} = 1$, otherwise, $D_{ip} = 0$,$i \in I,p \in P$
$G_{ii'p}$	If MP p is used by workpiece$i$and then workpiecei',$G_{ii'p} = 1$, otherwise, $G_{ii'p} = 0$,$i \in I,i' \in I,p \in P$
X_{ijm}	If the operation $O_{ij}$is processed on machine m, $X_{ijm} = 1$, otherwise, $X_{ijm} = 0$; if $m \notin M_{ij}$,$X_{ijm} = 0$
$Y_{iji'j'm}$	If operation$O_{ij}$is processed on machine$m$before operation$O_{i'j'}$, $Y_{iji'j'm} =$ 1otherwise,$Y_{iji'j'm} = 0$
A_{ijf}	If operation O_{ij} is clamped by FP f,$A_{ijf} = 1$, otherwise, $A_{ijf} = 0$;if $f \notin F_{ij}$,$A_{ijf} = 0$
$Z_{iji'j'f}$	If FPf is used by operation$O_{ij}$and then operation$O_{i'j'}$,$Z_{iji'j'f} = 1$, otherwise, $Z_{iji'j'f} = 0$
ILF_{ijn}	If operationO_{ij} is loaded at STn,$ILF_{ijn} = 1$, otherwise, $ILF_{ijn} = 0$
IUF_{ijn}	If operationO_{ij} is unloaded at STn,$IUF_{ijn} = 1$, otherwise, $IUF_{ijn} = 0$
$LU_{iji'j'n}$	If operation$O_{ij}$is loaded before operation$O_{i'j'}$ is unloaded at ST n,$LU_{iji'j'n} = 1$, otherwise, $LU_{iji'j'n} = 0$
$LL_{iji'j'n}$	If operation$O_{ij}$is loaded before operation$O_{i'j'}$ is loaded at ST n,$LL_{iji'j'n} =$ 1, otherwise, $LL_{iji'j'n} = 0$
$UL_{iji'j'n}$	If operation $O_{ij}$is unloaded before operation$O_{i'j'}$ is loaded at ST n,$UL_{iji'j'n} = 1$, otherwise, $UL_{iji'j'n} = 0$
$UU_{iji'j'n}$	If operation$O_{ij}$is unloaded before operation$O_{i'j'}$ is unloaded at ST n,$UU_{iji'j'n} = 1$, otherwise, $UU_{iji'j'n} = 0$

5) ST resources are considered, making queuing for loading and unloading necessary.
6) LUT is considered, while transportation time is neglected due to the high-speed stacker crane.

The optimal scheduling plan of the small case is shown in Fig. 5, which illustrates the time occupied by operations on resources. In each block, the upper number denotes the workpiece index, while the lower number signifies the operation sequence.LF_frepresents that the operation is loaded onto F_f, UF_fmeans that the operation is unloaded from F_f, and WL_fshows that the operation waits to be loaded ontoF_f.

2.3. Mathematical modelling of FPASFM

Table 4 shows the notations used in the model, including parameters, time variables, and decision variables. The objective is to minimize the makespan, with key constraints shown in Sections 2.3.2 and 2.3.3. The complete model and the runnable GUROBI code to facilitate verification of the model's correctness are provided in the Appendix A.

The assumptions of the model are outlined below:

1) When time= 0, all resources are available, and raw workpieces have been stored onto MPs.
2) When processing, storing, and transporting workpieces, they need to be placed onto FPs or MPs.
3) The combination of fixtures and pallets remains unchanged within a certain period.
4) The loading and unloading time needs to be considered, and it varies for different FPs.
5) Once a workpiece finishes processing, a new raw workpiece can be inputted.
6) Processing is continuous, and the breakdown is not allowed.

2.3.1. Objective function

The objective function is to minimize makespan, as formulated in the Eq. (1).

$$\min C_{\max} \tag{1}$$

2.3.2. Resource constraints

2.3.2.1. Machine. Eqs. (2) and (3) ensure that each operation is exclusively processed on a single machine, and each machine undertakes only one operation at a time.

$$\sum\nolimits_{m=1}^{M} X_{ijm} = 1, \forall i \in I, \forall j \in J_i \tag{2}$$

$$C_{ij} \le S_{i'j'} + L(2 - X_{ijm} - X_{i'j'm}) + L(1 - Y_{iji'j'm}),\ \forall i \in I, \forall j \in J_i, \forall i' \in I, \forall j' \in J_{i'}, \forall m \in M \tag{3}$$

2.3.2.2. Fixture-pallet. Eqs. (4) and (5) guarantee that each operation is only allocated to an FP, and each FP is utilized for only one operation at a time.

$$\sum\nolimits_{f=1}^{F} A_{ijf} = 1, \forall i \in I, \forall j \in J_i \tag{4}$$

$$CUF_{ij} \le SLF_{i'j'} + L \times (2 - A_{ijf} - A_{i'j'f}) + L \times (1 - Z_{iji'j'f}),\ \forall i \in I, \forall j \in J_i, \forall i' \in I, \forall j' \in J_{i'}, \forall f \in F \tag{5}$$

2.3.2.3. Material-pallet. Eqs. (6) and (7) limit that each workpiece can be assigned to an MP, and each MP can place only one workpiece.

$$\sum\nolimits_{p=1}^{P} D_{ip} = 1, \forall i \in I \tag{6}$$

$$CUF_{ij} \le SWL_{i'j'} + L \times (2 - D_{ip} - D_{i'p}) + L \times (1 - G_{ii'p}),\ \forall i \in I, \forall i' \in I, \forall p \in P, j = |J_i|, j' = 1 \tag{7}$$

2.3.2.4. Setup station. Eqs. (8)-(11) calculate the variables that indicate whether the operation needs to be loaded or unloaded. Eqs. (12)-(15) restrict each setup station can only load or unload one operation at a time.

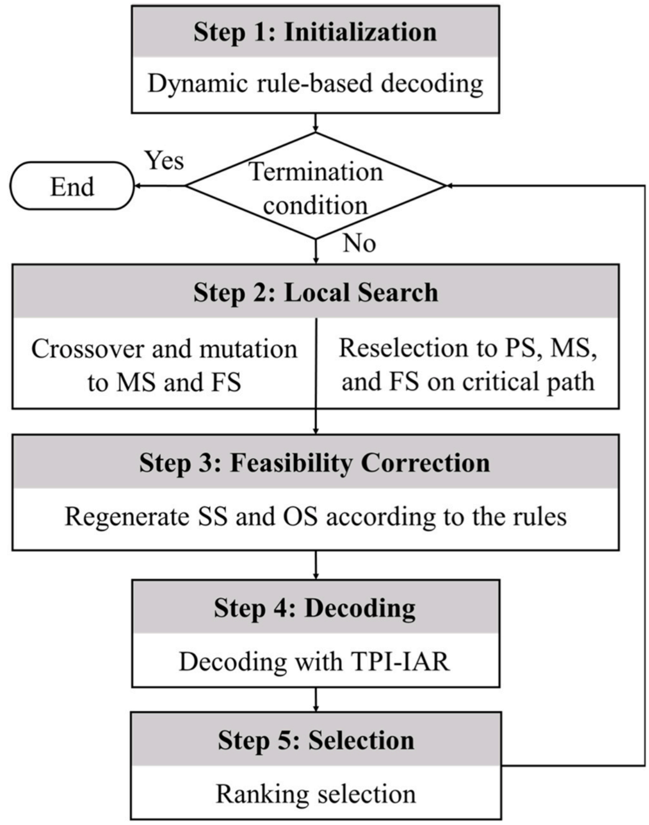

Fig. 6. The overview of IMHRC.

$$\sum\nolimits_{n=1}^{N} ILF_{ijn} = 1, \forall i \in I, j = 1 \tag{8}$$

$$\sum\nolimits_{n=1}^{N} ILF_{ijn} = \sum\nolimits_{f=1}^{F} (A_{ijf} \times (1 - A_{i(j-1)f})), \forall i \in I, j = 2, 3......|J_i| \tag{9}$$

$$\sum\nolimits_{n=1}^{N} IUF_{ijn} = 1, \forall i \in I, j = |J_i| \tag{10}$$

$$\sum\nolimits_{n=1}^{N} IUF_{ijn} = \sum\nolimits_{f=1}^{F} (A_{ijf} \times (1 - A_{i(j+1)f})), \forall i \in I, j = 1, 2......(|J_i| - 1) \tag{11}$$

$$CUF_{ij} \le SLF_{i'j'} + L(2 - IUF_{ijn} - ILF_{i'j'n}) + L(1 - UL_{iji'j'n}),\ \forall i \in I, \forall j \in J_i, \forall i' \in I, \forall j' \in J_{i'}, \forall n \in N \tag{12}$$

$$CLF_{ij} \le SLF_{i'j'} + L(2 - ILF_{ijn} - ILF_{i'j'n}) + L(1 - LL_{iji'j'n}),\ \forall i \in I, \forall j \in J_i, \forall i' \in I, \forall j' \in J_{i'}, \forall n \in N \tag{13}$$

$$CUF_{ij} \le SUF_{i'j'} + L(2 - IUF_{ijn} - IUF_{i'j'n}) + L(1 - UU_{iji'j'n}),\ \forall i \in I, \forall j \in J_i, \forall i' \in I, \forall j' \in J_{i'}, \forall n \in N \tag{14}$$

$$CLF_{ij} \le SUF_{i'j'} + L(2 - ILF_{ijn} - IUF_{i'j'n}) + L(1 - LU_{iji'j'n}),\ \forall i \in I, \forall j \in J_i, \forall i' \in I, \forall j' \in J_{i'}, \forall n \in N \tag{15}$$

2.3.3. Time constraints

Eqs. (16)-(20) collectively constrain the fundamental processing procedure of each workpiece, which must be performed in the order of waiting for loading, loading, processing, and unloading.

$$SWL_{ij} \le CWL_{ij}, \forall i \in I, j = 1, 2....|J_i| \tag{16}$$

$$CWL_{ij} \le SLF_{ij}, \forall i \in I, j = 1, 2....|J_i| \tag{17}$$

$$CLF_{ij} \le S_{ij}, \forall i \in I, j = 1, 2....|J_i| \tag{18}$$

$$C_{ij} \le SUF_{ij}, \forall i \in I, j = 1, 2....|J_i| \tag{19}$$

$$CUF_{i(j-1)} \le SWL_{ij}, \forall i \in I, j = 2......|J_i| \tag{20}$$

Eqs. (21)-(23) determine the loading, unloading, and processing time of each operation, respectively.

$$SLF_{ij} + \sum_{n=1}^{N} ILF_{ijn} \times \sum_{f=1}^{F} (A_{ijf} \times E_f) \le CLF_{ij}, \quad \forall i \in I, j = 1, 2...|J_i| \tag{21}$$

$$SUF_{ij} + \sum_{n=1}^{N} IUF_{ijn} \times \sum_{f=1}^{F} (A_{ijf} \times E_f) \le CUF_{ij}, \quad \forall i \in I, j = 1, 2, ...|J_i| \tag{22}$$

$$S_{ij} + \sum_{m=1}^{M} (T_{ijm} \times X_{ijm}) \le C_{ij}, \forall i \in I, \forall j \in J_i \tag{23}$$

Eq. (24) guarantees that the makespan exceeds the completion time for every individual operation.

$$C_{ij} \le C_{\max}, \forall i \in I, \forall j \in J_i \tag{24}$$

3. The proposed algorithm

3.1. The framework of the proposed algorithm

To address the complex problem presented in this paper, an improved meta-heuristic algorithm with rule-based initialization and critical path mutation (IMHRC) is developed. The overview of IMHRC is shown in Fig. 6, and the main five steps are outlined as follows:

Step 1: Initialization

A rule-based initialization method is applied, where resource selection and time arrangement are conducted alternately.

Step 2: Local search

Two local search methods are employed: crossover and mutation to MS and FS, and reselection to PS, MS, and FS on critical paths.

Step 3: Feasibility correction

Regenerate SS and OS according to the rules.

Step 4: Decoding

The decoding method with time period insertion based on the intersection of the available time of multiple resources (TPI-IAR) is applied.

Step 5: Selection

Using ranking selection, the new generation is chosen from the initial individuals.

The termination condition is that the global optimal solution remains unchanged for a certain number of consecutive generations, or the maximum iteration limit is reached.

3.2. Initialization

In this phase, a rule-based decoding approach is adopted, in which resource selection and time arrangement are conducted alternately as follows:

Step 1: Workpiece Sequencing (WS)

MTO (most total operations): sort the workpieces in decreasing order of the sum of total operations, i.e. $|J_i|$

MFFO (most alternative fixtures for the first operation): sort the workpieces in decreasing order of the number of alternative fixtures for the first operation, i.e. $|F_{i1}|$

STMPT (shortest total minimum processing time): sort the workpieces in the non-decreasing order of the sum of minimum processing time on alternative machines, i.e. $\sum_{j=1}^{|J_i|} \min_{m \in M_{ij}} T_{ijm}$

Step 2: Operation Sequencing (OS).

Sequence the operations according to the order of the workpieces.

Step 3: MP selection (PS)

EAT (earliest available time): select the MP for each workpiece with the earliest available time, i.e., select MP $p*$ such that $p* = \text{argmin}_{p \in P}\{CUF_{i|J_i|}\}$ when $D_{ip} = 1$.

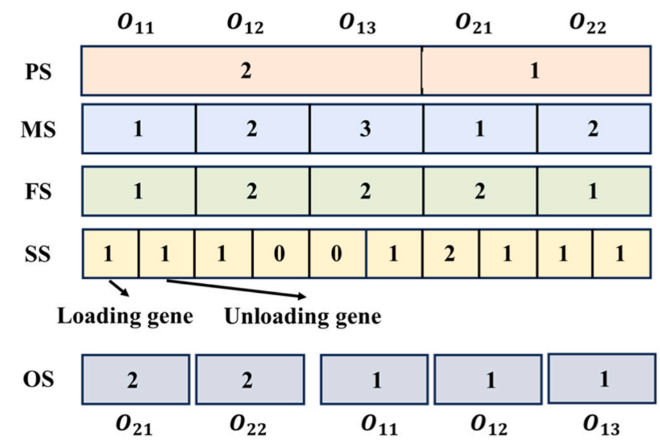

Fig. 7. The five-layer encoding scheme.

Step 4: FP selection (FS)

MNLU (minimum the number of loading and unloading): select the FP that enables the same FP to be used for more consecutive adjacent operations of the same workpiece, i.e., select FP $f*$ such that $f* = \text{argmax}_{f \in F_{ij}}\{\sum_{j=1}^{|J_i|-1} Z_{iji(j+1)f}\}$.

SLUT (shortest loading or unloading time): select the fixture with the shortest loading or unloading time, i.e., select f^* such that $f* = \text{argmin}_{f \in F_{ij}}\{E_f\}$.

EAT (earliest available time): select the fixture with the earliest available time, i.e., select $f*$ such that $f* = \text{argmin}_{f \in F_{ij}}\{\text{CUF}_{ij}\}$ when $A_{ijf} = 1$.

MLUL (minimum loading and unloading load): select the fixture with the smallest loading and unloading load, i.e., select fixture $f*$ such that $f* = \text{argmin}_{f \in F_{ij}}\{\sum_{i=1}^{i'} \sum_{j=1}^{j'} (ILF_{ijn} \times E_f + IUF_{ijn} \times E_f)\}$, when $A_{ijf} = 1$.

Step 5: ST Selection (SS) for loading.

EAT (earliest available time): select the ST with the earliest available time, i.e., select $n*$ such that $n* = \text{argmin}_{n \in N}\{IUF_{ijn} \times \text{CUF}_{ij}, ILF_{ijn} \times \text{CLF}_{ij}\}$

Step 6: Loading Time Arrangement.

$SLF_{ij} = \max$ {the end unloading time of the operation $O_{i(j-1)}$, the start time of the interval that both FP $f*$ and ST $n*$ are available, the previous end unloading time of FP $f*$, the earliest available time of MP $p*$}

$$CLF_{ij} = SLF_{ij} + E_f.$$

Step 7: Machine Selection (MS).

SPT (shortest processing time): select the machine with the shortest processing time, i.e., select $m*$ such that $m* = \text{argmin}_{m \in M_{ij}}\{T_{ijm}\}$

Step 8: Processing Time Arrangement.

$S_{ij} = \max$ {the end loading time of the operation O_{ij}, the start time of the interval that both machine $m*$ and FP $f*$ are available}

Step 9: ST Selection (SS) for unloading.

The step is the same as step 5.

Step 10: Unloading Time Arrangement.

$SUF_{ij} = \max$ {the end processing time of the operation O_{ij}, the start time of the interval that both FP $f*$ and ST $n*$ are available}

$$CUF_{ij} = SUF_{ij} + E_f.$$

Step 11: Update the Occupation Time of FP.

The occupation time of FP $f*$ is from SLF_{ij} to CUF_{ij}.

Step 12: Update the Occupation Time of MP.

The occupation time of FP $f*$ is from 0 to CUF_{ij} for the first operation and from $CUF_{i(j-1)}$ to CUF_{ij} for others.

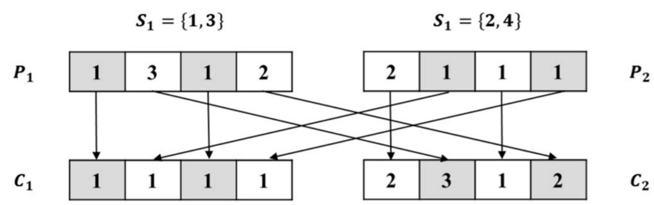

Fig. 8. Uniform Crossover.

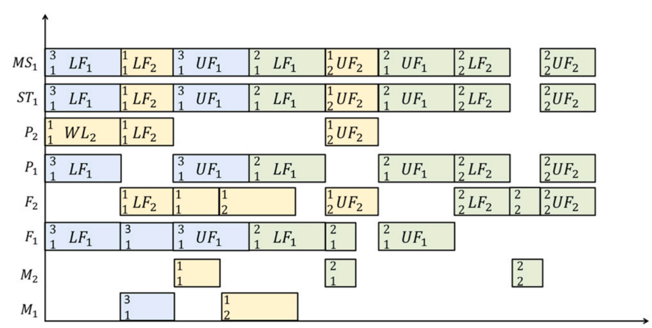

Fig. 9. A small case.

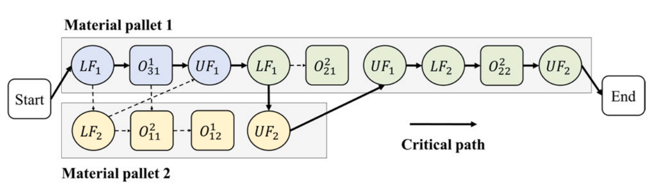

Fig. 10. The illustration for the critical paths.

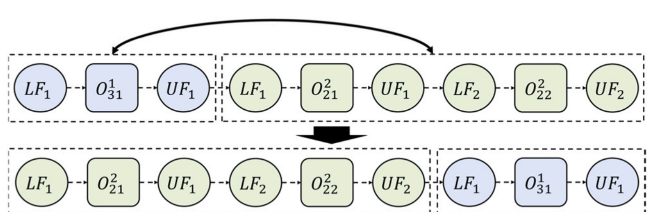

Fig. 11. Reselect PS.

Step 13: Repeat steps 3–12 until all operations are scheduled.

***Step 14: Record the result of resource selection in the form of* Fig. 7.**

The numbers in PS indicate the MP selection for each workpiece. The numbers in MS and FS represent the index of the machine and FP selected by the corresponding operation. Non-zero numbers in SS indicate the ST selection, and zero means no loading or unloading due to the same FP used by the adjacent operations. Note that each operation has two genes in SS, corresponding to loading and unloading, respectively.

3.3. Local search

Two methods are applied for local search.

1) A uniform crossover is initially performed (as illustrated in Fig. 8), followed by a multi-point mutation on the MS and FS of individuals with a fixed proportion.
2) For each individual selected according to a fixed proportion, a critical path is identified, and the PS, FS, and MS are reselected accordingly.

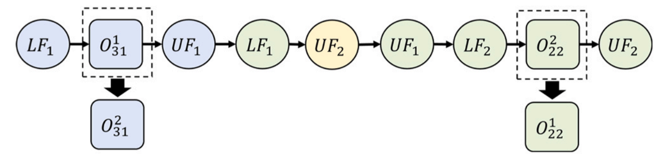

Fig. 12. Reselect MS.

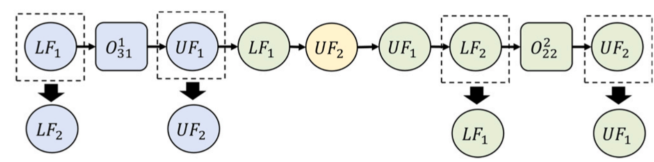

Fig. 13. Reselect FS.

The critical path for the small case in Fig. 9 is illustrated in Fig. 10, where O_{ij}^{m} represents operation O_{ij} is processed on the machine m. When the end time of one block equals the start time of the next block, they are connected by a dashed line; otherwise, there is no line. Identify critical paths that start from zero, end at the makespan, and are connected by dashed lines throughout. The solid line represents a feasible critical path.

Three different methods are applied to disrupt the chosen critical path.

1) Reselect PS

   In Fig. 9, workpiece 1 uses P_1 before workpiece 2. Exchange the order of using MP as shown in Fig. 11.
2) Reselect MS

   Reselect machines from alternative machine sets for partial O_{ij}^{m} blocks (see Fig. 12).
3) Reselect FS

Reselect fixtures from alternative fixture sets for partial LF_f or UF_f blocks (see Fig. 13).

3.4. Feasibility correction

In the encoding strategy, the SS relies on FS, while OS is grouped and disrupted based on SS. Consequently, if FS is altered but SS and OS remain unchanged, encoding errors will occur. To address this, feasibility correction can regenerate SS and OS before decoding, which was described in the paper [55].

3.5. Decoding

TPI-IAR consists of two stages: reading time periods and finding available time intervals, which were explained in detail in the paper [55]. Note that operations need to select the earliest available MP, and all operations of the same workpiece can only use the same MP. The MP will not be released and used by other workpieces until the production of the current workpiece is finished. The workpiece needs to be unloaded from the MP and then loaded onto the FP. Therefore, for each operation, it is necessary to ensure that the MP, FP, and ST are all available during loading and unloading.

3.6. Selection

Ranking selection is adopted when selecting the new better generation population, preserving individuals whose ranking is within the range of the initial population number.

Fig. 14. The PAS in the engine manufacturing enterprise.

Table 5
Time Complexity Estimation for Each Phase of the Proposed Algorithm.

Step	Description	Complexity
Initialization	Resource selection and time arrangement	$O(P\times n\times r)$
Local search	Crossover and mutation; Critical path-based mutation strategy	$O(P\times (n^2+n\times r))$
Feasibility correction	Regenerate SS and OS sequences	$O(P\times n)$
Decoding	Reading time periods and finding available time intervals	$O(P\times n\times r)$
Selection	Ranking selection	$O(P\times \log P)$

Table 6
15 cases generated from the real data.

Scale	Case	W	O	M	FP	MP	ST	MS
Small	1	3	3	2	2	3	1	1
	2	3	3	3	2	3	1	1
	3	5	3	5	5	5	1	1
	4	8	6	6	5	5	1	1
	5	10	5	5	6	4	1	1
Medium	6	10	10	8	8	12	2	2
	7	10	12	8	8	12	2	2
	8	12	15	10	10	10	3	3
	9	12	15	12	10	10	3	3
	10	15	10	12	12	8	3	3
Large	11	20	10	12	15	15	4	4
	12	20	14	12	15	15	4	4
	13	30	15	15	18	22	5	5
	14	40	15	15	18	22	5	5
	15	50	9	18	20	20	5	5

3.7. IMHRC complexity analysis

The purpose of this section is to analyze the computational complexity of the proposed algorithm, focusing on its key steps. This analysis helps to evaluate the algorithm's scalability and efficiency under different problem sizes. Let the following parameters be defined:

1) P: population size
2) n: total number of operations
3) r: total number of flexible resources, including machines (m), fixture-pallets (f), material-pallets (p), and setup stations (s)

The algorithm consists of five major steps. The estimated complexity of each step is summarized in Table 5.

Given that the total number of resources r is usually less than or close to n, the overall complexity after L iterations can be simplified as:

$$O\left(P\times L\times n^2\right)$$

4. Case study

4.1. Experimental settings

4.1.1. Data sources

The real data is sourced from one of the largest Chinese engine manufacturing enterprises. The workshop employs several PASs, one of which can simultaneously accommodate 30 MPs and FPs (Fig. 14). This system is equipped with 5 machines and 2 STs. Currently, the workshop relies on manual scheduling, resulting in low resource utilization and a high rate of order delay.

4.1.2. Case design

15 cases are generated from the real data, which consists of three scales, as listed in Table 6. The columns in “W”, “O”, “M”, “FP”, “MP”, “ST”, and “MS” represent the number of workpieces, operations, machines, fixture-pallets, material-pallets, setup stations, and material stations, respectively.

4.1.3. Parameter setting

To determine the optimal parameter settings for the proposed algorithm, a series of experiments is conducted based on Case 3 to evaluate how different parameter values affect solution quality. Each setting is tested over 10 independent runs, and the average makespan is used for analysis. When varying one parameter, all other parameters are kept constant to ensure a controlled comparison.

Fig. 15 illustrates the effect of five key parameters—Population Size, Max Iterations, Crossover Rate, Mutation Rate, and Critical Path Rate—on makespan.

Based on the observed trends in the figures, the optimal values for each parameter are summarized in Table 7. These selected values are used as the default configuration for the algorithm in the subsequent experiments.

4.2. Study on rules for initialization

To explore the performance of the distinct rule combinations, 100 experiments for each case across each rule combination are conducted. The rules for selecting resources include workpiece sequencing (MTO, MFFO, and STMPT), machine selection (SPT, MPL, and EAT), FP selection (MNLU, SLUT, EAT, and MLUL), MP selection (EAT), and ST selection (EAT), where minimum processing load (MPL) means selecting the machine with the smallest processing load. It is denoted as the selection of the first rule for all five resources as "1–1-1–1-1", representing MTO-SPT-MBLU-EAT-EAT. Therefore, there are a total of 36 rule combinations.

To standardize the makespan from different cases into the same range, this paper employs a Min-Max Normalization (MMN) method, as shown in the Eq. (29), to scale the data within the range of [0, 1]. Here, μ_i represents the average of 100 solutions obtained from solving Case i using one kind of rule combination. $\min(\mu_i)$ denotes the minimum value among the 36 rule combinations for solving Case i, while $\max(\mu_i)$ represents the maximum value.

$$MMN=\frac{\mu_i-\min(\mu_i)}{\max(\mu_i)-\min(\mu_i)} \tag{29}$$

To further evaluate the average gap of each rule in identifying relatively optimal solutions (IRO), Eq. (30) is utilized.

$$IRO=\sum\nolimits_{i=1}^{15}\frac{\mu_i-\min(\mu_i)}{\min(\mu_i)} \tag{30}$$

Fig. 16-Fig. 18 present the MMN and IRO comparisons of different rule combinations across various cases. The meaningful and useful insights are clarified as follows:

Insight 1. *The rule combinations MTO-SPT-MBLU-EAT-EAT, MFFO-*

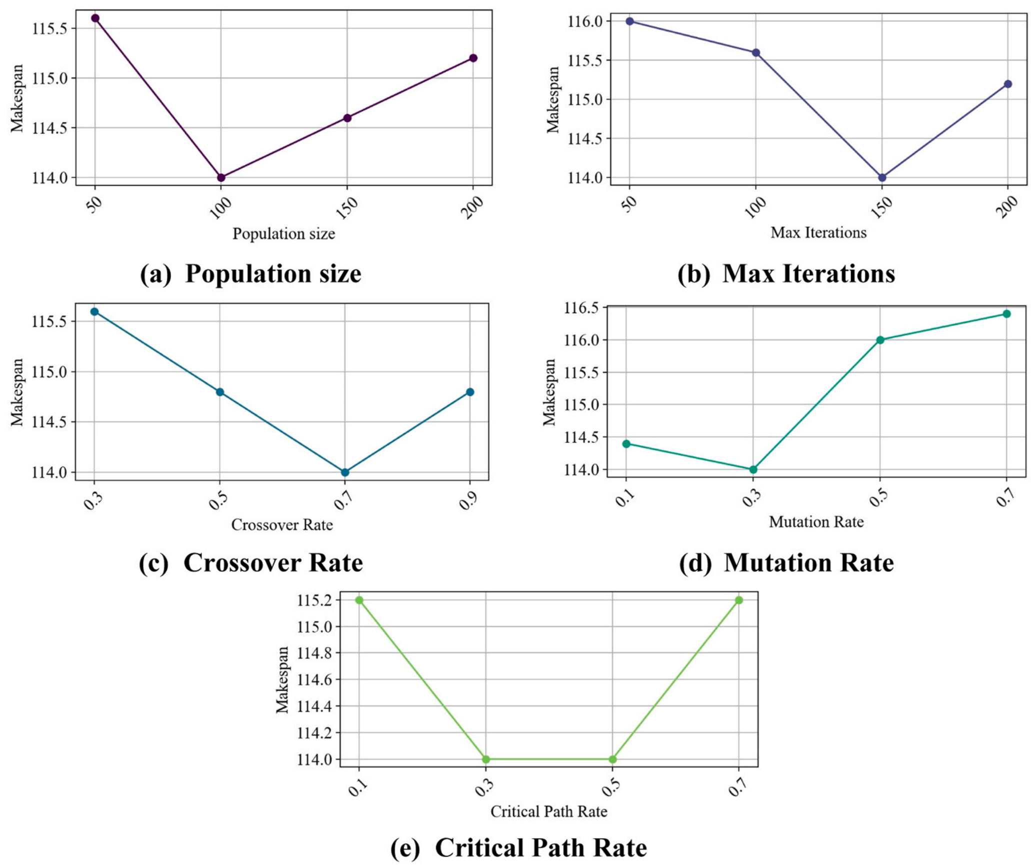

Fig. 15. The effect of five parameters on makespan.

Table 7
The optimal values for each parameter.

Parameter	Tested Values	Best Value	Average Makespan
Population size	50, 100, 150, 200	100	114
Max Iterations	50, 100, 150, 200	150	114
Crossover Rate	0.9, 0.7, 0.5, 0.3	0.7	114
Mutation Rate	0.7, 0.5, 0.3, 0.1	0.3	114
Critical Path Rate	0.7, 0.5, 0.3, 0.1	0.3	114

SPT-MBLU-EAT-EAT, and STMPT-SPT-MBLU-EAT-EAT demonstrate strong performance, with STMPT-SPT-MBLU-EAT-EAT being the best. (from Fig. 18)

Insight 1.1. *For workpiece sequencing, STMPT rule slightly performs better than MTO and MFFO rules.* (from Fig. 18)

Insight 1.2. *For machine selection, SPT rule significantly outperforms MPL and EAT rules.* (from Fig. 17 and Fig. 18)

Insight 1.3. *For FP selection, MNLU rule performs the best, though SLUT, EAT, and MLUL rules also have a certain probability of obtaining the optimal solution.* (from Fig. 17 and Fig. 18)

Additionally, the performance of different rules shows randomness in Cases 1–5 (Fig. 16). Therefore, to obtain a better initial solution during initialization, the three workpiece sequencing rules are all applied, and the machine selection rule is fixed to the SPT rule. For fixture selection, the MNLU rule is primarily used, with occasional use of the other three rules for randomness. When only the final solution is generated using rules—rather than the initial one—*STMPT-SPT-MBLU-EAT-EAT* demonstrates the best performance.

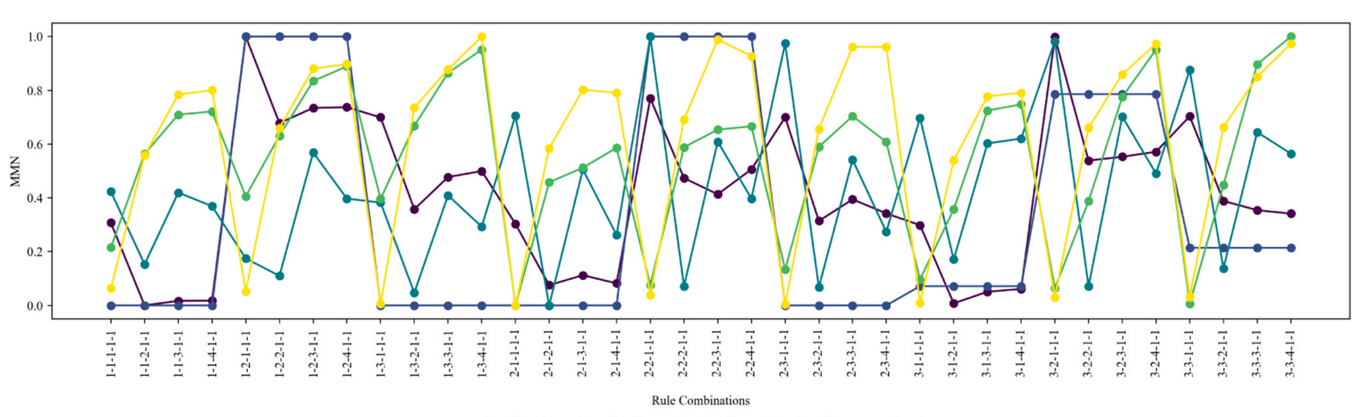

Fig. 16. MMN comparison of different rule combinations on Case 1 to Case 5.

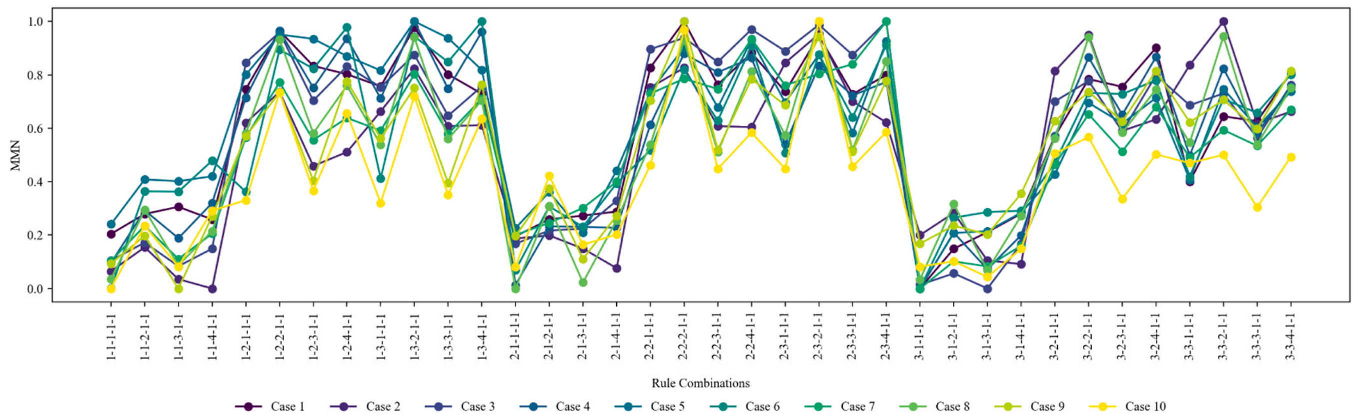

Fig. 17. MMN comparison of different rule combinations on Case 6 to Case 15.

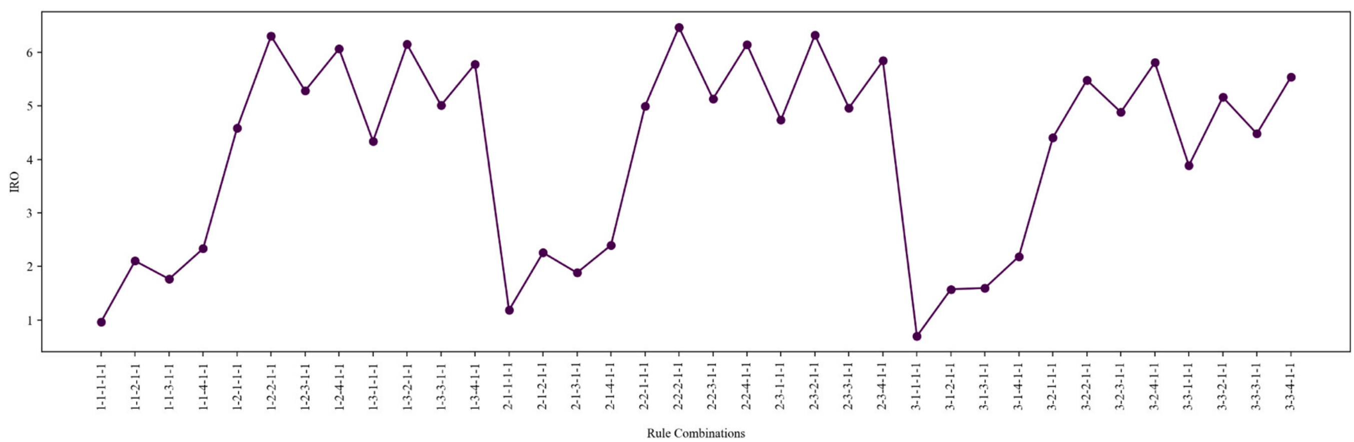

Fig. 18. IRO comparison of different rule combinations on Cases 1–15.

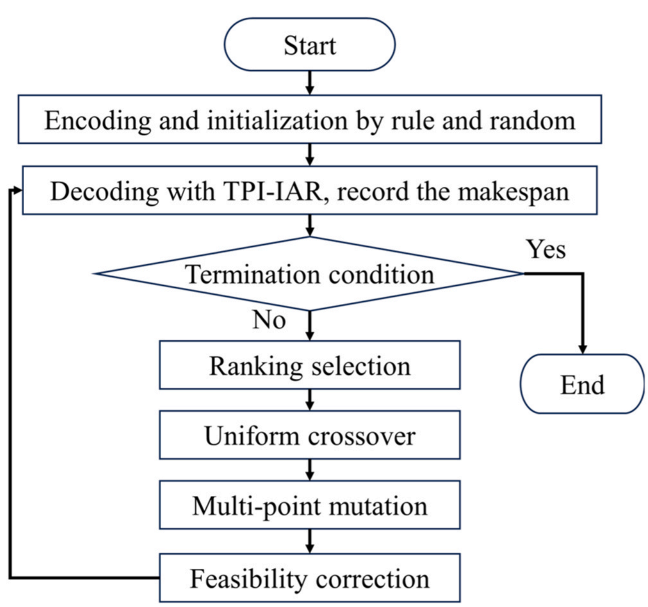

Fig. 19. The flowchart of GA.

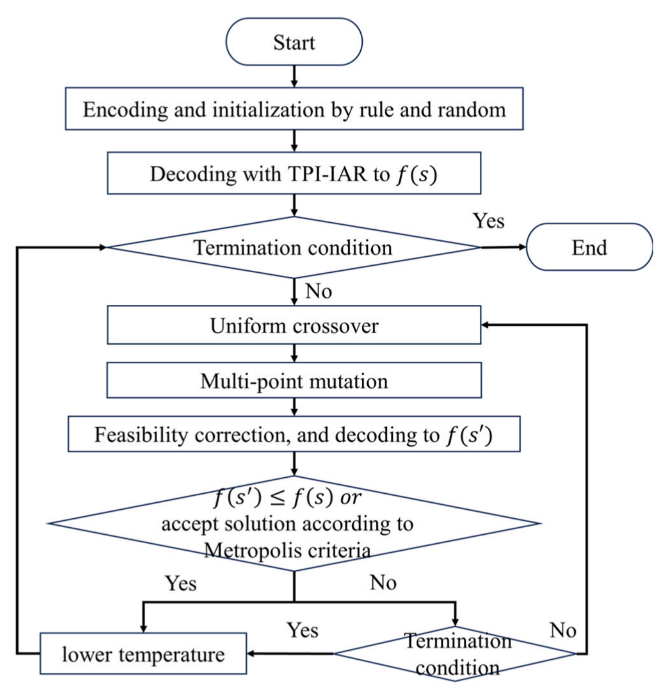

Fig. 20. The flowchart of SA.

4.3. Performance of IMHRC

To evaluate the accuracy of the mathematical model and the performance of IMHRC, it is benchmarked against GUROBI, genetic

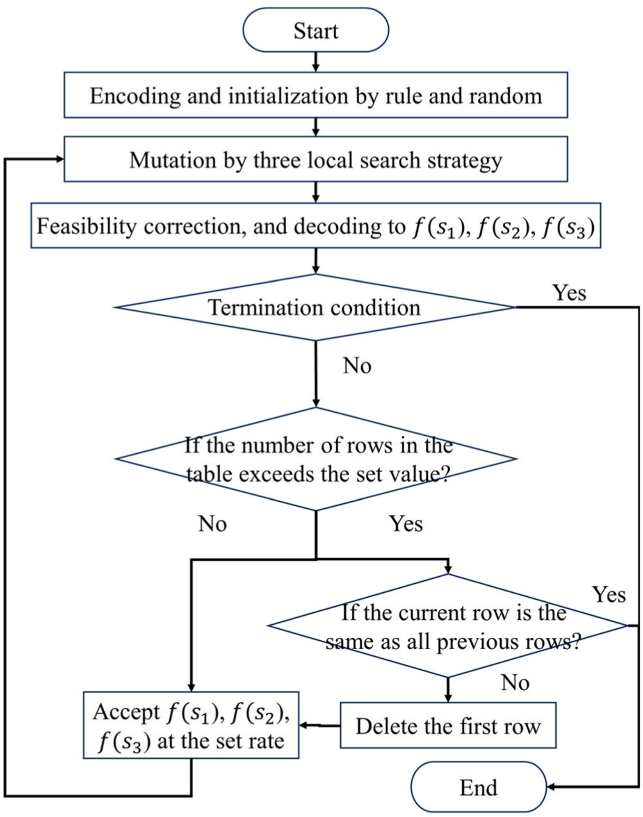

Fig. 21. The flowchart of TS.

algorithm (GA), simulated annealing (SA), tabu search (TS), hyper-heuristic (HH), improved meta-heuristics with rule-based initialization (IMHR), and improved meta-heuristics with critical path mutation (IMHC). For the GUROBI mathematical programming solver, time limits are set to 3600 s, 7200 s, and 14400 s for small-, medium-, and large-scale cases, respectively. The applied HH consists of three phases. In the first phase, GA is employed to perform a global search and generate well-structured initial solutions. The second phase applies SA to perturb and refine the solution, enhancing the ability to escape from local optima. Finally, a greedy search is used to further optimize the solution locally by accepting only improving moves, aiming to accelerate convergence and improve solution quality.

In GA, the population size, maximum iterations, crossover rate, mutation rate, and termination condition are the same as IMHRC. For SA, the number of iterations is set to 10,000, the limit on consecutive generations is 2000, the initial temperature is 200, and the neighborhood size is 10. In TS, the number of iterations is also set to 10,000, while the tabu list size is restricted to 5 rows. In HH, the iterations L is set to 300. All these parameters were determined based on a series of experimental evaluations. The flowcharts of GA, SA, and TS are illustrated in Fig. 19, Fig. 20, and Fig. 21, respectively.

The solution effectiveness of these algorithms is assessed using metric Q_i, defined by Eq. (31). Here, $\overline{Opt_1}$ represents the optimal solution obtained by GUROBI within the given time constraints. Meanwhile, $\overline{Opt_2}$,$\overline{Opt_3}$,$\overline{Opt_4}$, $\overline{Opt_5}$,$\overline{Opt_6}$,$\overline{Opt_7}$, and $\overline{Opt_8}$ denote the average values derived from 10 runs of GA, SA, TS, HH, IMHR, IMHC, and IMHRC, respectively. When Q_i is positive, it indicates that IMHRC performs better than others. The larger value of these metrics, the larger difference in the performance gap between algorithms.$\overline{T_i}(i = 1, 2, 3, 4, 5, 6, 7, 8)$represent the average time required to solve the cases.

$$Q_i = \frac{\overline{Opt_i} - \overline{Opt_8}}{\overline{Opt_i}} \times 100\%, i = 1, 2, 3, 4, 5, 6, 7 \tag{31}$$

Table 8
Comparison of the solution time and the optimal value of the eight algorithms.

	Case	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Gurobi	$\overline{T_1}$	1.0	0.3	3600	3600	3600	7200	7200	7200	7200	7200	14400	14400	14400	14400	14400
	$\overline{Opt_1}$	57.0	76.0	112.0	-	-	-	-	-	-	-	-	-	-	-	-
GA	$\overline{T_2}$	4.5	5.3	11.8	76.1	120.8	287.3	473.5	871.1	747.1	306.0	472.1	805.6	1961.6	3651.0	1588.1
	$\overline{Opt_2}$	57.0	76.0	116.6	292.2	777.8	368.6	588.8	1059.8	1102.0	758.2	773.8	1576.6	2448.0	3464.8	1537.6
SA	$\overline{T_3}$	1.1	0.8	1.9	19.5	29.6	120.0	210.2	385.0	783.9	358.3	742.3	2426.2	3342.2	5544.1	6019.1
	$\overline{Opt_3}$	62.4	76.0	173.8	441.2	1096.2	493.6	772.0	1282.0	1301.6	906.6	909.0	1659.6	2741.0	3808.6	1666.8
TS	$\overline{T_4}$	0.4	2.2	8.4	83.6	84.0	163.6	386.0	1173.3	681.2	497.5	883.7	1214.7	3947.5	3973.2	2654.4
	$\overline{Opt_4}$	59.4	76.0	127.8	349.0	852.4	461.2	682.6	1231.6	1222.8	847.8	895.4	1765.8	2669.4	3709.8	1646.0
HH	$\overline{T_5}$	5.4	7.0	11.9	70.3	79.8	213.1	396.5	760.0	671.1	330.4	435.9	796.0	1850.1	3489.2	1567.1
	$\overline{Opt_5}$	57.6	76.0	116.0	292.2	779.8	353.0	574.2	1093.2	1104.8	773.2	767.4	1518.6	2328.4	3269.3	1490.3
IMHR	$\overline{T_6}$	2.4	3.2	7.4	46.6	138.2	153.3	245.3	373.1	399.6	333.0	488.3	980.4	2054.6	3107.2	1555.2
	$\overline{Opt_6}$	57.0	76.0	115.6	291.3	770.2	313.5	547.2	894.5	935.6	669.7	672.0	1442.3	2236.0	2945.0	1390.8
IMHC	$\overline{T_7}$	3.7	4.2	11.2	73.0	134.6	240.9	390.7	442.9	407.1	334.1	520.3	1348.5	2539.2	3309.4	1809.2
	$\overline{Opt_7}$	57.0	76.0	115.0	291.5	775.4	330.6	568.3	995.2	971.1	691.0	699.0	1495.0	2368.0	3098.3	1415.2
IMHRC	$\overline{T_8}$	3.1	3.7	8.7	52.3	92.5	115.4	190.1	368.9	285.9	216.6	384.0	701.5	1518.6	2839.4	1519.0
	$\overline{Opt_8}$	57.0	76.0	114.0	287.2	765.8	291.0	484.4	860.4	827.4	582.8	633.0	1236.4	1765.8	2589.2	1178.0
Q_1		0.00 %	0.00 %	−1.79 %	-	-	-	-	-	-	-	-	-	-	-	-
Q_2		0.00 %	0.00 %	2.23 %	1.71 %	1.54 %	21.05 %	17.73 %	18.81 %	24.92 %	23.13 %	18.20 %	21.58 %	27.87 %	25.27 %	23.39 %
Q_3		8.65 %	0.00 %	34.41 %	34.90 %	30.14 %	41.05 %	37.25 %	32.89 %	36.43 %	35.72 %	30.36 %	25.50 %	35.58 %	32.02 %	29.33 %
Q_4		4.04 %	0.00 %	10.80 %	17.71 %	10.16 %	36.90 %	29.04 %	30.14 %	32.34 %	31.26 %	29.31 %	29.98 %	33.85 %	30.21 %	28.43 %
Q_5		1.04 %	0.00 %	1.72 %	1.71 %	1.80 %	17.56 %	15.64 %	21.30 %	25.11 %	24.62 %	17.51 %	18.58 %	24.16 %	20.80 %	20.96 %
Q_6		0.00 %	0.00 %	1.38 %	1.41 %	0.57 %	7.18 %	11.48 %	3.81 %	11.56 %	12.98 %	5.80 %	14.28 %	21.03 %	12.08 %	15.30 %
Q_7		0.00 %	0.00 %	0.87 %	1.48 %	1.24 %	11.98 %	14.76 %	13.55 %	14.80 %	15.66 %	9.44 %	17.30 %	25.43 %	16.43 %	16.76 %

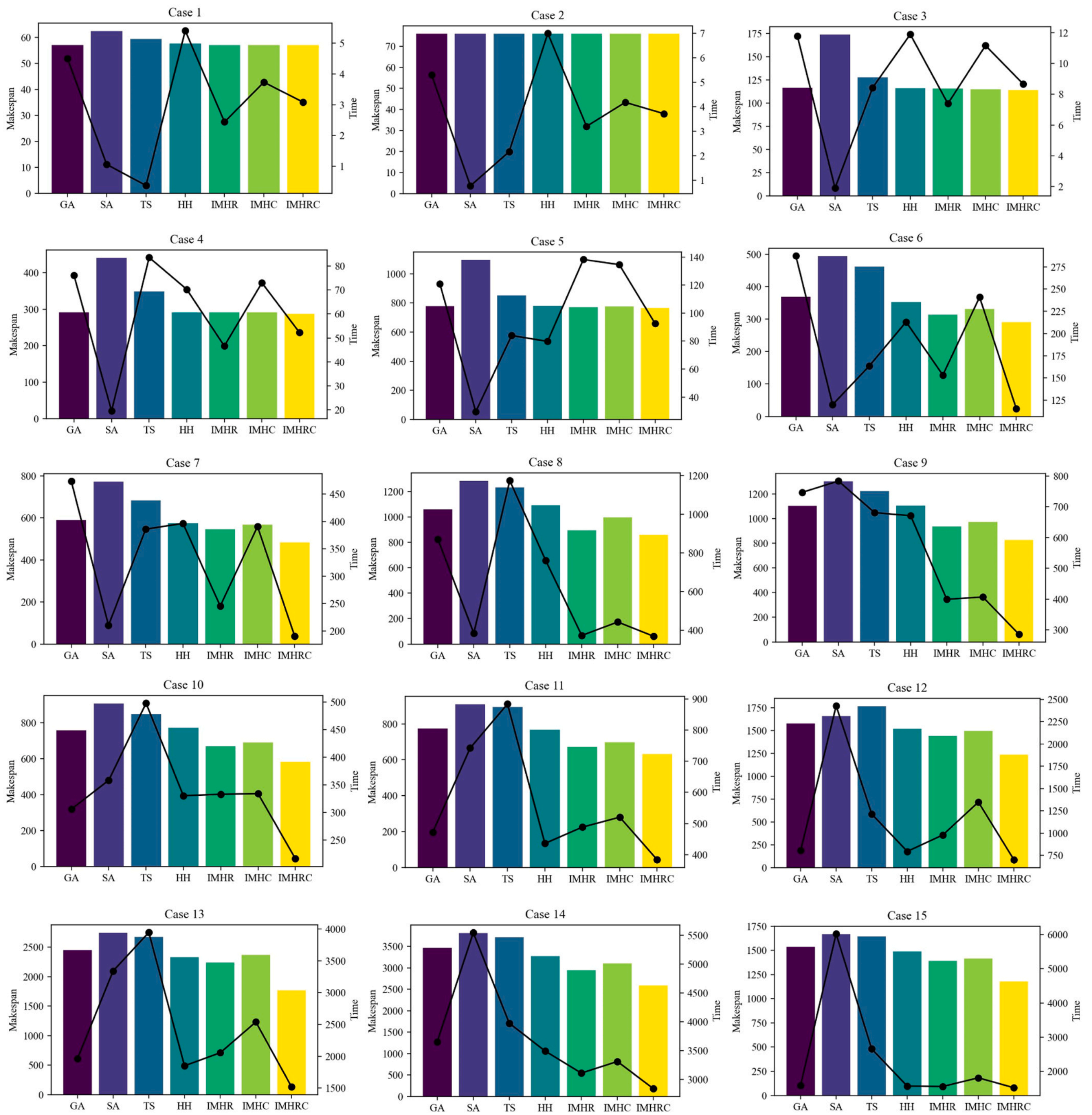

Fig. 22. The comparative results of makespan and time across 15 cases.

Table 8 presents a comparison of the average solution time and optimal values obtained by the eight algorithms across 15 cases. The comparative results of makespan and time across 15 cases are illustrated in Fig. 22, with insights as follows.

Insight 2. *IMHRC can achieve better solutions in shorter times compared to other algorithms in most cases.*

Insight 2.1. *The accuracy of the model, encoding strategy, and decoding method are validated by Case 1 and Case 2. In Case 2, all algorithms, including GUROBI, reach the same optimal solution of 100. Similarly, GUROBI, GA, IMHR, IMHC, and IMHRC all achieve the exact optimal solution of 57 in Case 1.*

Insight 2.2. *Compared with the six algorithms GA, SA, TS, HH, IMHR, and IMHC, IMHRC outperforms them in metric Q_i by 15.16 %, 29.61 %, 23.61 %, 14.17 %, 7.92 %, and 10.65 %, respectively. Therefore, the ranking of solution effectiveness among these six algorithms is: IMHRC > IMHR > IMHC >HH> GA > TS > SA.*

Insight 2.3. *The rule-based initialization contributes more significantly to performance improvement than the critical path-based local search.*

Insight 2.4. *In medium- and large-scale instances, IMHRC's superior performance in solution time highlights the effectiveness of its initialization and local search strategies.*

Insight 2.5. *The complexity of the problem arises with the increasing number of resources, making it impossible for GUROBI to find a feasible solution within the given time for Case 3 to Case 15.*

Table 9
Comparison of the results of manual scheduling and IMHRC.

Order scale	Method	Makespan	Average machine utilization	Average FP utilization	Average ST utilization	Average MP utilization
Small	Manual scheduling	233	17.43 %	34.93 %	62.66 %	61.72 %
	IMHRC	94	40.43 %	61.06 %	78.72 %	79.15 %
Medium	Manual scheduling	671	21.55 %	35.44 %	26.83 %	84.65 %
	IMHRC	363	35.59 %	56.31 %	33.89 %	87.40 %

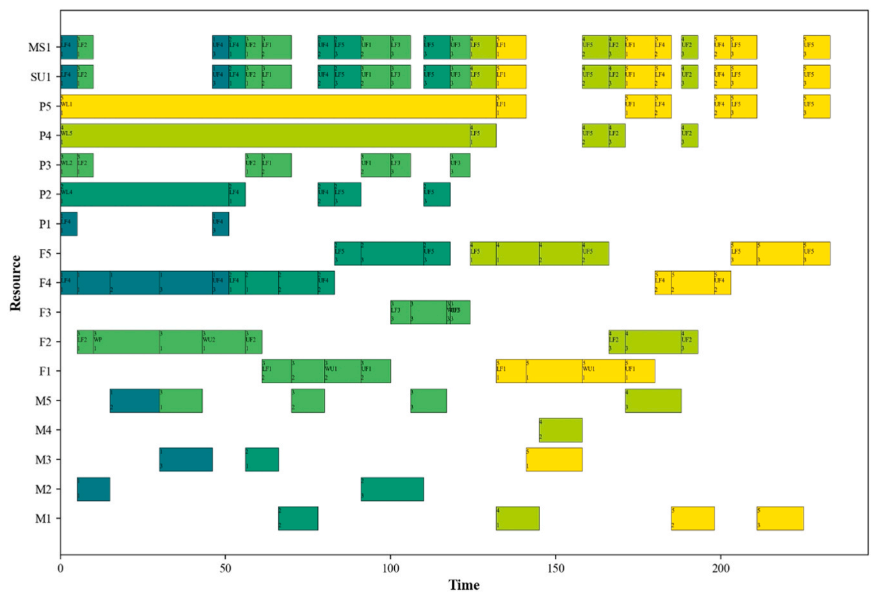

Fig. 23. The Gantt chart of small order by manual scheduling.

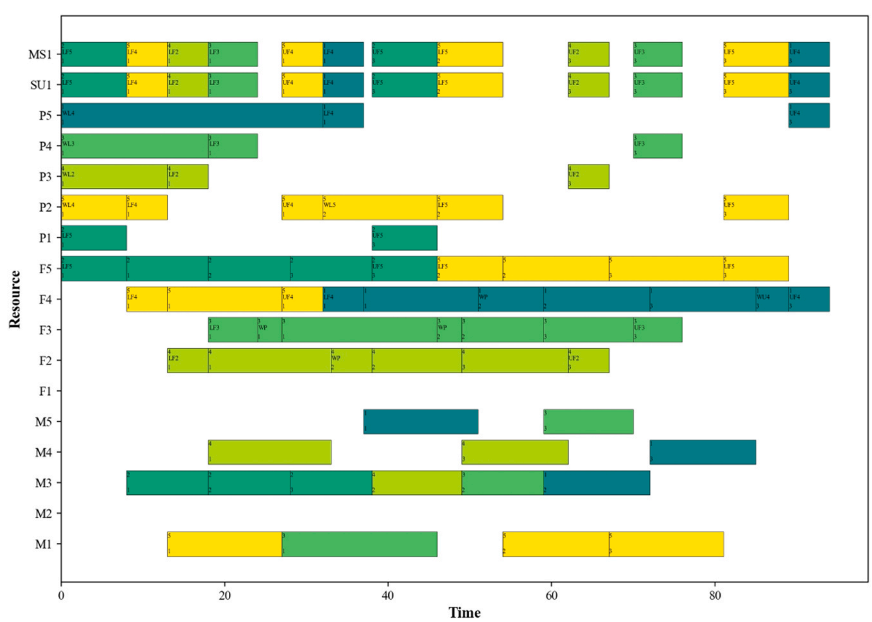

Fig. 24. The Gantt chart of small order by IMHRC.

4.4. Improvements in real orders

In the actual production of this enterprise, orders typically involve multiple products and are used mostly for prototype manufacturing. For example, consider two orders: a small order with 5 products and a medium order with 10 products. Both orders are processed in the PAS equipped with 5 machines, 5 FPs, and 5 MPs. For the medium order, the number of MPs is insufficient to accommodate all products. Therefore, only up to 5 products can be processed simultaneously in the first batch, while the remaining products await the availability of MPs.

Previously, the enterprise adopted manual scheduling, prioritizing the production of products with the most complex processes and selecting the earliest available machine, fixture, and pallet for processing.

Table 9 presents a comparative analysis of the results for small and medium orders, contrasting manual scheduling with IMHRC. The evaluated metrics include makespan, average machine utilization, average FP utilization, average ST utilization, and average MP utilization. Fig. 23 and Fig. 24 depict the Gantt charts for the small order scheduled manually and by IMHRC, respectively. The insights are listed as follows.

Insight 3. *IMHRC holds high application value in actual order production*

Insight 3.1. *IMHRC achieves a substantial reduction in makespan, by 59.66 % and 45.90 % for the two orders, respectively.*

Insight 3.2. *IMHRC significantly improves all utilization metrics.*

5. Conclusions and future work

PAS serves as a key solution for enhancing flexible automation in enterprises. It enables efficient storage, organization, and scheduling of various high-flexibility limited resources, significantly reducing non-processing time and improving resource utilization. Within the system, FPs and MPs are two critical resources. Workpieces need to be clamped on FPs for processing, causing different operations to compete for limited FPs, while MPs store raw materials and semi-finished workpieces to release FPs' capacity and limit the number of workpieces that can be produced simultaneously in PAS. However, the allocation challenges of these two resources complicate the scheduling plans.

To address this, the study investigates FPASFM aimed at minimizing makespan. We establish a mathematical model and propose IMHRC, which integrates a five-layer encoding scheme and a TPI-IAR decoding method. It employs efficient rules such as SPT and MNLU to generate high-quality initial solutions and incorporates three operators based on critical paths to enhance local search capabilities. Firstly, the experiments on 36 rule combinations identified optimal initialization rules. Furthermore, in 15 cases generated from real data, IMHRC achieves superior solutions compared to other algorithms within a shorter computational time, with performance ranking as follows: IMHRC > IMHR > IMHC > HH> GA > TS > SA. Additionally, in two real-world order cases, IMHRC significantly reduces makespan by 59.66 % and 45.90 %, respectively, while markedly improving resource utilization rates.

To address the limitations of the current study, future research directions include:

1) Exploring the impact of additional resources (e.g., buffer resources) on PAS scheduling.
2) Developing effective rescheduling methods to handle dynamic challenges such as urgent resource shortages.
3) Investigating the potential of reinforcement learning techniques in solving MRFJSP.

These directions are expected to promote the development of more intelligent, flexible, and responsive scheduling strategies for multi-resource-constrained environments.

CRediT authorship contribution statement

Yulu Zhou: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Siyang Wang:** Resources, Project administration. **Andrea Matta:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Xiaoxiao Shen:** Writing – review & editing, Data curation. **Jun Lv:** Visualization, Validation. **Shichang Du:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by National Natural Science Foundation of China (Grant No. 52275499 and No. 92467101) and by China Scholarship Council's International Cooperation Training Project for Innovative Talents. This study was also carried out within the Motown: Smart Production Planning and Control for Manufacturing of Electric Vehicle Powertrain in Industry 4.0 Environment project – funded by European Union – Next Generation EU within the PRIN 2022 program (D.D. 104 - 02/02/2022 Ministero dell' Università e della Ricerca). This manuscript reflects only the authors' views and opinions and the Ministry cannot be considered responsible for them.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.jmsy.2025.09.015.

Data Availability

The datasets used in this study (Cases 1–15), together with the runnable GUROBI code implementing the proposed scheduling model, are openly available in the Zenodo repository at https://doi.org/10.5281/zenodo.15464152.

References

[1] Gania IP, Stachowiak A, Oleśków-Szłapka J. FLEXIBLE MANUFACTURING SYSTEMS: INDUSTRY 4.0 SOLUTION. DEStech Trans Eng Technol Res 2018. https://doi.org/10.12783/dtetr/icpr2017/17583.

[2] Li L, Jiang Y. Fastems White paper on pallet automation application in China. FASTEMS 2018. Available from: ⟨https://www.fastemschina.com⟩.

[3] Zhou Y, Du S, Liu M, Shen X. Machine-fixture-pallet resources constrained flexible job shop scheduling considering loading and unloading times under pallet automation system. J Manuf Syst 2024;73:143–58. https://doi.org/10.1016/j.jmsy.2024.01.010.

[4] Dhiflaoui M, Nouri HE, Driss OB. Dual-Resource constraints in classical and flexible job shop problems: a State-of-the-Art review. Procedia Comput Sci 2018;126:1507–15. https://doi.org/10.1016/j.procs.2018.08.123.

[5] Li J, Han Y, Gao K, Xiao X, Duan P. Bi-population balancing multi-objective algorithm for fuzzy flexible job shop with energy and transportation. IEEE Trans Autom Sci Eng 2024;21:4686–702. https://doi.org/10.1109/TASE.2023.3300922.

[6] Pan Z, Wang L, Wang J, Yu Y, Li R. Distributed energy-efficient flexible manufacturing with assembly and transportation: a knowledge-based bi-hierarchical optimization approach. IEEE Trans Autom Sci Eng 2024:1–17. https://doi.org/10.1109/TASE.2024.3396474.

[7] Li R, Gong W, Wang L, Lu C, Dong C. Co-Evolution with deep reinforcement learning for Energy-Aware distributed heterogeneous flexible job shop scheduling. IEEE Trans Syst Man Cyber Syst 2024;54:201–11. https://doi.org/10.1109/TSMC.2023.3305541.

[8] Wei S, Tang H, Li X, Lei D, Wang XV. An improved memetic algorithm for multi-objective resource-constrained flexible job shop inverse scheduling problem: an application for machining workshop. J Manuf Syst 2024;74:264–90. https://doi.org/10.1016/j.jmsy.2024.03.005.

[9] Perrachon Q, Olteanu A-L, Sevaux M, Fréchengues S, Kerviche J-F. Industrial multi-resource flexible job shop scheduling with partially necessary resources. Eur J Oper Res 2025;320:309–27. https://doi.org/10.1016/j.ejor.2024.07.023.

[10] Wei Z., Yu Z., Niu R., Zhao Q., Li Z. Research on flexible job shop scheduling method for agricultural equipment considering multi-resource constraints 2025. https://doi.org/10.20944/preprints202501.1439.v1.

[11] Kasapidis GA, Paraskevopoulos DC, Mourtos I, Repoussis PP. A unified solution framework for flexible job shop scheduling problems with multiple resource constraints. Eur J Oper Res 2025;320:479–95. https://doi.org/10.1016/j.ejor.2024.08.010.

[12] Chen C-A, Kuo H-A, Chien C-F. Dual-resource constrained flexible job shop scheduling for SMT back-end production and an empirical study of wearable devices. IEEE Trans Autom Sci Eng 2024:1–10. https://doi.org/10.1109/TASE.2024.3503412.

[13] Saber RG, Leyman P, Aghezzaf E-H. Modeling the integrated flexible job-shop and operator scheduling in flexible manufacturing systems. IFACPap 2024;58:343–8. https://doi.org/10.1016/j.ifacol.2024.09.235.

[14] Luo X, Guo S, Du B, Luo X, Guo J. Multi-objective multi-skill resource-constrained project scheduling considering flexible resource profiles. Appl Sci 2024;14:1921. https://doi.org/10.3390/app14051921.

[15] Fan D, Wang C. A genetic algorithm for the dual resource constrained flexible job shop scheduling problem considering preparation times. 2024 12th Int. Conf. Traffic Logist. Eng. ICTLE. Macau, China: IEEE; 2024. p. 128–32. https://doi.org/10.1109/ICTLE62418.2024.10703949.
[16] Gao Z, Feng Y, Xing K. A hybrid Estimation-of-Distribution algorithm for scheduling flexible job shop with limited buffers based on petri nets. IEEE Access 2020;8:165396–408. https://doi.org/10.1109/ACCESS.2020.3022655.
[17] Chang D, Shi H, Han C, Meng F. Research on production scheduling optimization of flexible job shop production with buffer capacity limitation based on the improved gene expression programming algorithm. Int J Precis Eng Manuf 2023;24:2317–36. https://doi.org/10.1007/s12541-023-00897-2.
[18] Homayuni SM, Fontes DBMM, Fontes FACC. A hybrid BRKGA for joint scheduling production, transport, and Storage/Retrieval in flexible job shops. Proc. Companion Conf. Genet. Evol. Comput. Lisbon Portugal: ACM; 2023. p. 211–4. https://doi.org/10.1145/3583133.3590758.
[19] Hu M, Chen H, Luo H, Zhang J, Wang S. An improved genetic algorithm for flexible job shop scheduling problem with buffer constraints. 2021 26th Int. Conf. Autom. Comput. ICAC. Portsmouth, United Kingdom: IEEE; 2021. p. 1–7. https://doi.org/10.23919/ICAC50006.2021.9594239.
[20] Liao X, Zhang R, Chen Y, Song S. A new artificial bee colony algorithm for the flexible job shop scheduling problem with extra resource constraints in numeric control centers. Expert Syst Appl 2024. https://doi.org/10.1016/j.eswa.2024.123556.
[21] Zhou Y-D, Shin J-H, Lee D-H. Loading and scheduling for flexible manufacturing systems with controllable processing times. Eng Optim 2019;51:412–26. https://doi.org/10.1080/0305215X.2018.1469134.
[22] Liu Q, Gao Z, Li J, Li S, Zhu L. Research on optimization of dual-resource batch scheduling in flexible job shop. Comput Mater Contin 2023;76:2503–30. https://doi.org/10.32604/cmc.2023.040505.
[23] Yao Y, Liu Q, Fu L, Li X, Yu Y, Gao L, et al. A novel mathematical model for the flexible job-shop scheduling problem with limited automated guided vehicles. IEEE Trans Autom Sci Eng 2024:1–14. https://doi.org/10.1109/TASE.2024.3356255.
[24] Wang D, Peng X, Qin H. Improved genetic algorithm for optimizing flexible workshop scheduling. In: Zhang K, Lorenz P, editors. Int. Conf. Mechatron. Intell. Control ICMIC. Wuhan, China: SPIE; 2024–25. p. 180. https://doi.org/10.1117/12.3051800.
[25] Ahn J, Kim H-J. A branch and bound algorithm for scheduling of flexible manufacturing systems. IEEE Trans Autom Sci Eng 2024;21:4382–96. https://doi.org/10.1109/TASE.2023.3296087.
[26] Zhu X, Xu J, Ge J, Wang Y, Xie Z. Multi-Task Multi-Agent reinforcement learning for Real-Time scheduling of a Dual-Resource flexible job shop with robots. Processes 2023;11:267. https://doi.org/10.3390/pr11010267.
[27] Liao Q, Yang J, Zhou Y. Sustainable scheduling of an automatic pallet changer system by Multi-Objective evolutionary algorithm with first piece inspection. Sustainability 2019;11:1498. https://doi.org/10.3390/su11051498.
[28] Mati Y, Lahlou C, Dauzère-Pérès S. Modelling and solving a practical flexible job-shop scheduling problem with blocking constraints. Int J Prod Res 2011;49:2169–82. https://doi.org/10.1080/00207541003733775.
[29] Diabat A, Araidah OA, Alsyouf I, Duh C. A heuristic approach to scheduling jobs in machining centres equipped with automated pallet changers. Int J Adv Oper Manag 2011;3:326. https://doi.org/10.1504/IJAOM.2011.045447.
[30] Li J, Li X, Gao L, Liu Q. A flexible job shop scheduling problem considering On-Site machining fixtures: a case study from customized manufacturing enterprise. IEEE Trans Autom Sci Eng 2024:1–12. https://doi.org/10.1109/TASE.2024.3485810.
[31] Li J, Liu Q, Wang C, Li X. A disjunctive graph-based metaheuristic for flexible job-shop scheduling problems considering fixture shortages in customized manufacturing systems. Robot ComputIntegr Manuf 2025;95:102981. https://doi.org/10.1016/j.rcim.2025.102981.
[32] Chen JC, Chen T-L, Chen Y-Y, Chung M-Y. Multi-resource constrained scheduling considering process plan flexibility and lot streaming for the CNC machining industry. Flex Serv Manuf J 2024;36:946–93. https://doi.org/10.1007/s10696-023-09514-w.
[33] Cheng Y, Xie Z, Xin Y, Chen K, Zarei R. Flexible job shop scheduling method for optimizing mold resource setup time. IEEE Access 2024;12:33486–503. https://doi.org/10.1109/ACCESS.2024.3372396.
[34] Yunusoglu P, Topaloglu Yildiz S. Solving the flexible job shop scheduling and lot streaming problem with setup and transport resource constraints. Int J Syst Sci Oper Logist 2023;10:2221072. https://doi.org/10.1080/23302674.2023.2221072.
[35] Wu X, Peng J, Xiao X, Wu S. An effective approach for the dual-resource flexible job shop scheduling problem considering loading and unloading. J Intell Manuf 2021;32:707–28. https://doi.org/10.1007/s10845-020-01697-5.
[36] Liu M, Lv J, Du S, Deng Y, Shen X, Zhou Y. Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimisation. Comput Ind Eng 2024;188:109903. https://doi.org/10.1016/j.cie.2024.109903.
[37] Sim D-G, Lee D-H. A two-stage solution approach for scheduling flexible manufacturing systems with multi-fixturing pallets. Flex Serv Manuf J 2024. https://doi.org/10.1007/s10696-024-09567-5.
[38] Shin J-H, Yu J-M, Doh H-H, Kim H-W, Lee D-H. Batching and scheduling for a single-machine flexible machining cell with multi-fixturing pallets and controllable processing times. Int J Prod Res 2020;58:863–77. https://doi.org/10.1080/00207543.2019.1602742.
[39] Lee D-K, Shin J-H, Lee D-H. Operations scheduling for an advanced flexible manufacturing system with multi-fixturing pallets. Comput Ind Eng 2020;144:106496. https://doi.org/10.1016/j.cie.2020.106496.
[40] Yu J-M, Doh H-H, Kim J-S, Kwon Y-J, Lee D-H, Nam S-H. Input sequencing and scheduling for a reconfigurable manufacturing system with a limited number of fixtures. Int J Adv Manuf Technol 2013;67:157–69. https://doi.org/10.1007/s00170-013-4761-z.
[41] Hajibabaei M, Behnamian J. Fuzzy cleaner production in assembly flexible job-shop scheduling with machine breakdown and batch transportation: lagrangian relaxation. J Comb Optim 2023;45:112. https://doi.org/10.1007/s10878-023-01046-1.
[42] Soto C, Dorronsoro B, Fraire H, Cruz-Reyes L, Gomez-Santillan C, Rangel N. Solving the multi-objective flexible job shop scheduling problem with a novel parallel branch and bound algorithm. Swarm Evol Comput 2020;53:100632. https://doi.org/10.1016/j.swevo.2019.100632.
[43] Juvin C, Houssin L, Lopez P. Logic-based benders decomposition for the preemptive flexible job-shop scheduling problem. Comput Oper Res 2023;152:106156. https://doi.org/10.1016/j.cor.2023.106156.
[44] Wang H, Peng T, Nassehi A, Tang R. A data-driven simulation-optimization framework for generating priority dispatching rules in dynamic job shop scheduling with uncertainties. J Manuf Syst 2023;70:288–308. https://doi.org/10.1016/j.jmsy.2023.08.001.
[45] Zhang L, Hu Y, Wang C, Tang Q, Li X. Effective dispatching rules mining based on near-optimal schedules in intelligent job shop environment. J Manuf Syst 2022;63:424–38. https://doi.org/10.1016/j.jmsy.2022.04.019.
[46] Xu Y, Zhang M, Yang M, Wang D. Hybrid quantum particle swarm optimization and variable neighborhood search for flexible job-shop scheduling problem. J Manuf Syst 2024;73:334–48. https://doi.org/10.1016/j.jmsy.2024.02.007.
[47] Schworm P, Wu X, Klar M, Glatt M, Aurich JC. Multi-objective quantum annealing approach for solving flexible job shop scheduling in manufacturing. J Manuf Syst 2024;72:142–53. https://doi.org/10.1016/j.jmsy.2023.11.015.
[48] Huang L, Tang D, Zhang Z, Zhu H, Cai Q, Zhao S. An iterated greedy algorithm integrating job insertion strategy for distributed job shop scheduling problems. J Manuf Syst 2024;77:746–63. https://doi.org/10.1016/j.jmsy.2024.10.014.
[49] Guo H, Liu J, Wang Y, Zhuang C. An improved genetic programming hyper-heuristic for the dynamic flexible job shop scheduling problem with reconfigurable manufacturing cells. J Manuf Syst 2024;74:252–63. https://doi.org/10.1016/j.jmsy.2024.03.009.
[50] Fan C, Wang W, Tian J. Flexible job shop scheduling with stochastic machine breakdowns by an improved tuna swarm optimization algorithm. J Manuf Syst 2024;74:180–97. https://doi.org/10.1016/j.jmsy.2024.03.002.
[51] Xie J, Li X, Gao L, Gui L. A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems. J Manuf Syst 2023;71:82–94. https://doi.org/10.1016/j.jmsy.2023.09.002.
[52] Li X, Guo A, Yin X, Tang H, Wu R, Zhao Q, et al. A Q-learning improved differential evolution algorithm for human-centric dynamic distributed flexible job shop scheduling problem. J Manuf Syst 2025;80:794–823. https://doi.org/10.1016/j.jmsy.2025.04.001.
[53] Ding L, Guan Z, Luo D, Yue L. Data-driven hierarchical multi-policy deep reinforcement learning framework for multi-objective multiplicity dynamic flexible job shop scheduling. J Manuf Syst 2025;80:536–62. https://doi.org/10.1016/j.jmsy.2025.03.019.
[54] Zhang W, Zhao F, Li Y, Du C, Feng X, Mei X. A novel collaborative agent reinforcement learning framework based on an attention mechanism and disjunctive graph embedding for flexible job shop scheduling problem. J Manuf Syst 2024;74:329–45. https://doi.org/10.1016/j.jmsy.2024.03.012.
[55] Zhou Y, Lv J, Du S, Shen X, Liu M. Multi-resource constrained flexible pallet automation system scheduling considering fixture-pallet and setup station. TechRxiv 2025. https://doi.org/10.36227/techrxiv.174439259.90707217/v1.