Multiresource-Constrained Flexible Job Shop Scheduling With Fixture Pallets and Setup Stations Under Pallet Automation Systems

Yulu Zhou, Jun Lv, Shichang Du, *Member, IEEE*, Xiaoxiao Shen, *Member, IEEE*, and Molin Liu

***Abstract*—Pallet automation systems (PASs) are critical in flexible manufacturing for their ability to integrate diverse production resources, such as machines, fixture pallets (FPs), and setup stations (STs). In traditional flexible manufacturing systems (FMSs), loading/unloading typically occurs at machines, limiting the machines' capacity. To address this, in PASs, workpieces are loaded and unloaded from the machines at a limited number of STs, so loading, processing, and unloading are three separate segments. However, the research on PASs is limited, and existing studies mainly focused on machines while overlooking FPs and STs. A critical challenge is the tight coupling between resource selection and operation sequencing. To address this gap, we investigate a multiresource-constrained flexible job shop scheduling problem (MRFJSP) with FPs and STs under PASs (MRFFS). First, a mixed-integer programming model is proposed to minimize makespan. Second, a four-layer encoding scheme and a new decoding method with time period insertion based on the intersection of available time of multiple resources (TPI-IARs) are presented to obtain feasible schedule solutions and shrink the search space. Third, a new search algorithm based on critical paths and points mutation (SACP) is developed to effectively balance exploration and exploitation. Finally, four case studies are designed to demonstrate the validity and effectiveness of this work.**

***Index Terms*—Critical paths and points, fixture, multiresource constraints, pallet automation system (PAS), setup station (ST).**

NOMENCLATURE

I	Workpiece set indexed by $i, i' = 1, 2, \ldots, \|I\|$.
J_i	Operation set of workpiece i indexed by $j, j' = 1, 2, \ldots, \|J_i\|$.
M	Machine total set indexed by m, $m = 1, 2, \ldots, \|M\|$.
M_{ij}	Available machine set of operation O_{ij}, $M_{ij} \in M$.
F	FP total set indexed by f, $f = 1, 2, \ldots, \|F\|$.
F_{ij}	Available FP set of operation O_{ij}, $F_{ij} \in F$.
O_{ij}	jth operation of workpiece i, $i \in I$, $j \in J_i$.
N	ST total set of the PAS indexed by n, $n = 1, 2, \ldots, \|N\|$.
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	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}.

0–1 Decision Variables

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 $O_{i''}$, $Y_{iji''}$; otherwise, $Y_{iji''}$.
A_{ijf}	If operation O_{ij} is fixed by FP f, $A_{ijf} = 1$; otherwise, $A_{ijf} = 0$; if $f \notin F_{ij}$, $A_{ijf} = 0$.
$Z_{iji'j'f}$	If FP f is used by operation O_{ij} and then operation $O_{i''}$, $Z_{iji''}$; otherwise, $Z_{iji''}$.
$UL_{iji'j'f}$	If operation O_{ij} is unloaded before operation $O_{i'j'}$ is loaded on FP f, $UL_{iji'j'f} = 1$; otherwise, $UL_{iji'j'f} = 0$.
ILF_{ijn}	If operation O_{ij} is loaded at ST n, $ILF_{ijn} = 1$; otherwise, $ILF_{ijn} = 0$.
IUF_{ijn}	If operation O_{ij} is unloaded at ST n, $IUF_{ijn} = 1$; 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, $UL_{iji'j'n} = 0$.

Received 11 June 2025; revised 21 August 2025 and 25 December 2025; accepted 13 January 2026. This work was supported by the National Natural Science Foundation of China under Grant 52275499 and Grant 92467101. This article was recommended by Associate Editor Q. Gou. *(Corresponding author: Shichang Du.)*

Yulu Zhou, Shichang Du, and Xiaoxiao Shen are with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yuluzhou@sjtu.edu.cn; lovbin@sjtu.edu.cn; sjtusxx98@sjtu.edu.cn).

Jun Lv is with the Faculty of Economics and Management, East China Normal University, Shanghai 200241, China (e-mail: jlv@dbm.ecnu.edu.cn).

Molin Liu is with the Technology and Operations Management Area, INSEAD, 77305 Fontainebleau, France (e-mail: molin.liu@insead.edu).

Color versions of one or more figures in this letter are available at https://doi.org/10.1109/TSMC.2026.3655483.

Digital Object Identifier 10.1109/TSMC.2026.3655483

$UU_{iji'j'n}$ If operation O_{ij} is unloaded before operation $O_{i'j'}$ is unloaded at ST n, $UU_{iji'j'n} = 1$; otherwise, $UU_{iji'j'n} = 0$.

I. Introduction

PALLET automation systems (PASs) have emerged in response to the need for increased production flexibility in the face of varying lot sizes and product types [1]. PASs integrate standardized pallets, computer-controlled machines, and material-handling systems to help enterprises store, organize, and schedule multiple limited resources, achieving high levels of automation and adaptability. Traditional flexible manufacturing systems (FMSs) only enable the rapid transfer of workpieces between different machines, with fixtures fixed on the machines. As a result, the loading and unloading of workpieces must be performed on the machines themselves. This causes the loading and unloading time (LUT) to occupy the machines' processing time, leading to significant capacity waste during nonprocessing operations [2].

In PAS, fixtures are fixed on pallets to form fixture pallets (FPs), and loading/unloading occurs off-machine at a limited number of setup stations (STs), which benefits a high utilization rate and maximizes the machines' output. Different operations require different alternative FPs during processing, necessitating loading and unloading with type-dependent time. In this case, FPs and STs are critical resources. FP selection limits the processing order and determines whether loading/unloading is required, while ST selection governs the loading/unloading queues and their timing. Therefore, FP selection determines whether an ST must be visited, while ST selection affects whether the FP can be released in time. This makes the coordinated scheduling and allocation of FPs and STs crucial for minimizing the makespan.

However, few studies have investigated the complex problem. Most studies on the flexible job shop scheduling problem (FJSP) simplify the problem by considering only machine constraints [3], [4], [5], [6]. In contrast, multiresource-constrained FJSP (MRFJSP) more closely reflects real-world production demands by incorporating resource constraints [7], [8], [9], such as tools [10], [11], workers [12], [13], [14], [15], robots [16], [17], [18], and AGVs [19], [20], [21], [22]. Nevertheless, studies on FPs and STs remain relatively scarce.

Table I summarizes the relevant works on MRFJSP about fixture, pallet, ST, and LUT constraints. Some researchers considered the integration of flexible machines and limited pallets [23], while others explored the availability of machines and fixtures [24], [25], [26], [27], [28], [29]. A few studies focused on machines, fixtures, and pallets simultaneously, but they overlooked some real-world production constraints. Liu et al. [30] fixed pallets on each machine, preventing fixtures from being shared between machines. Lee et al. [31], Shin et al. [32], Doh et al. [33], and Sim and Lee [34] neglected the flexibility of fixtures and the varying LUT for workpieces on different FPs. Zhou et al. [35] underscored the significance of the LUT for workpieces but ignored ST constraints. As far as we know, research on fixtures and pallets is limited, and no existing work has studied PAS with STs, which highlights a significant research gap in this area.

TABLE I
Summary of Research Articles in Key Resources

Literature	Method	Pallet	Fixture	LUT	Other constraints
[49]	M			√	
[23]	M	√		√	
[24]	M		√	√	Transport resource
[25]	M		√	√	
[26]	M		√	√	
[27]	M		√	√	
[31]	P	√	√		
[32]	P	√	√		
[33]	P	√	√		
[28]	M		√		
[30]	M	√	√		
[35]	M	√	√	√	
[34]	M*	√	√		
[29]	M*		√		
[45]	M*				Auxiliary modules
[48]	M*				Workers and auxiliary resources
Ours	M*	√	√	√	Setup station

M: Meta-heuristic
P: Priority rule
*With critical-path neighborhood search

To address MRFJSP, some methods that have been widely used for FJSP, such as meta-heuristic algorithms [36], [37], [38], priority rules [39], and machine learning tools [40], [41], [42], [43] have also been adopted in this area. As shown in Table I, research on the MRFJSP with fixture and pallet resources mainly applied meta-heuristic algorithms and priority rules.

Recent meta-heuristics increasingly strengthen local search using critical path neighborhoods, and some studies also allow resource-related changes during neighborhood moves. Classical scheduling theory indicates that the makespan equals the length of the longest path from the source to the sink (i.e., the critical path) [44]. Fan et al. [45] noted that if a neighborhood move does not change the critical path, the makespan will generally not be improved. Xie et al. [46] proposed N8, which allows moving critical operations outside their critical blocks to enlarge the search space. Liu et al. [47] further strengthened such moves, e.g., swapping a critical block operation with an operation on the noncritical path. Zhang et al. [48] argued that traditional neighborhoods are insufficient under multiresource constraints, and enabled neighborhood moves to change resource selections while avoiding resource conflicts. Under fixture resource constraint, Li et al. [29] incorporated fixture-usage ordering into critical path search and designed neighborhoods that jointly adjust operation sequencing and machine/fixture assignments.

However, the above studies mainly focus on processing activities, whereas loading and unloading are also nonnegligible in real production. Therefore, it is necessary to separate loading, processing, and unloading into distinct activities and include both processing and loading/unloading times in the makespan. Accordingly, the critical path should be extended to a time chain that includes loading/unloading events, based on which corresponding neighborhoods can be constructed. Moreover, as the number of resource types increases, conventional decoding methods may fail to capture the tight coupling between resource selection and operation sequencing, often generating infeasible schedules and thus increasing the computational burden.

Hence, a multiresource-constrained flexible job shop scheduling with FPs and STs under PASs (MRFFS) is studied in this article, which is an NP-hard problem [50]. It extends typical MRFJSP by explicitly modeling FPs and STs within PASs, introducing three key features: 1) queuing for loading/unloading occurs off-machine at a limited number of STs; 2) each operation selects among alternative FPs with FP-type-dependent times; and 3) whether loading/unloading is required depends on whether FP is kept or switched between consecutive operations. On the modeling side, capturing these mechanisms requires additional assignment and ordering binaries, continuous timing variables, and big-M logical linearization. On the algorithmic side, the challenge is to design an encoding and decoding scheme that captures the coupling between resource selection and operation sequencing to reduce infeasible solutions and develop local search algorithms that jointly perturb FP and machine to improve computational efficiency.

To overcome these challenges, the proposed methods and main contributions are described as follows.

1) A mixed-integer programming model that can define MRFFS is established. Special constraints determine FP assignment, whether loading/unloading occurs at STs, and the ordering of load/unload events, and they also ensure that an FP is not reused during the waiting period between loading and unloading.
2) A search algorithm based on critical paths and points mutation (SACP) is developed. A four-layer encoding scheme and a decoding method with time period insertion based on the intersection of available time of multiple resources (TPI-IARs) are designed to force adjacency of consecutive operations that require no loading/unloading, shrink the search space, and improve efficiency.
3) Four case studies indicate that SACP outperforms the other four algorithms. The critical path/point mutation is effective on two real orders. SACP reduces makespan and improves resource utilization relative to manual scheduling. In addition, the ST allocation study identifies the optimal number of STs for the production setting.

The structure of this article is outlined as follows. System description, problem definition, and modeling, including assumptions, objectives, and constraints, are provided in Section II. Section III introduces the proposed method to solve the problem. Section IV shows the experimental results through four case studies. Section V concludes this article and outlines potential directions for future research.

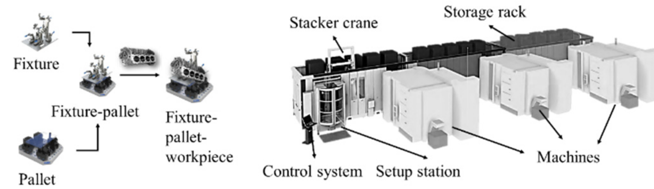


Fig. 1. Configuration of PAS.

TABLE II
2 WORKPIECES–3 MACHINES–2 FPs–2 STs (MIN)

Workpiece	Operation	Machine M_1	M_2	M_3	Alternative FP
1	O_{11}	5	6	5	F_1
	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
Alternative ST		ST_1, ST_2			

TABLE III
LOADING OR UNLOADING TIME OF A WORKPIECE ON THE FP

FP	Loading time or unloading Time
F_1	5
F_2	3

II. MODEL OF MRFFS

A. System Description

A PAS comprises several key components: a control system, machines with different functions, a storage rack for storing FPs or FP-workpieces (FPWs), a stacker crane with high moving speed, and STs (see Fig. 1).

As shown in Fig. 2, the PAS production process consists of four sequential stages.

1) *Production Preparation:* Fixtures are assembled onto pallets to form FPs.
2) *Load Workpieces:* Load workpieces onto FPs at STs to create FPWs.
3) *Process Workpieces:* FPWs are transported to machines via a stacker crane for processing.
4) *Unload Workpieces:* If the next operation uses the same FP, the FPW is transferred directly to the next machine. If a different FP is needed, the FPW returns to an ST for unloading, followed by reloading onto a new FP. Based on whether consecutive operations share the same FP, the loading/unloading requirements at STs are determined.

B. Problem Definition

The 2 workpieces–3 machines–2 FPs–2 STs are taken as an instance (see Table II) to clarify MRFFS. Each operation can be processed on multiple alternative machines with varying

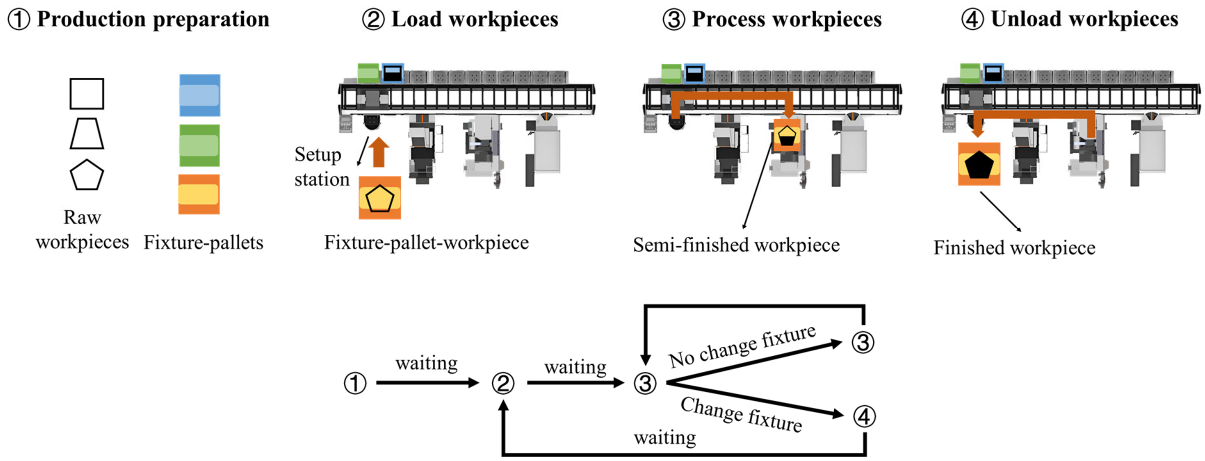


Fig. 2. Description of the production process in PAS.

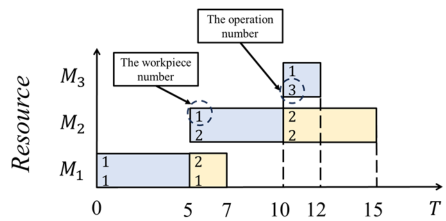


Fig. 3. Gantt chart of FJSP.

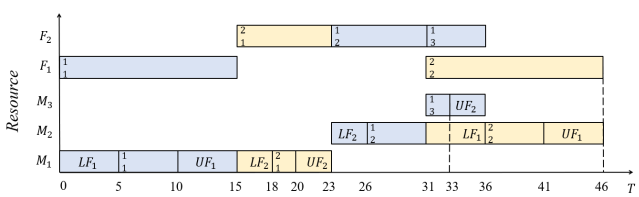


Fig. 4. Gantt chart of MRFJSP with FPs and LUT.

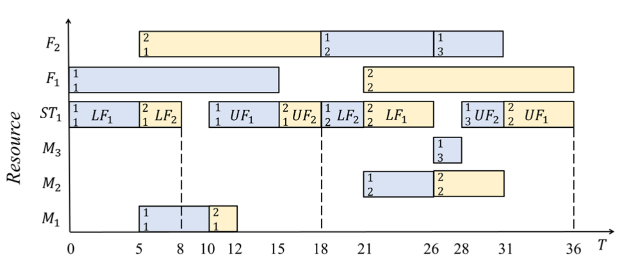


Fig. 5. Gantt chart of MRFJSP with FPs, an ST, and LUT.

processing times, allowing different alternative FPs whose LUT depends on the FPs' type (see Table III).

To illustrate the challenges of MRFFS studied in this article, the scheduling results of four MRFJSP with increasing complexity based on the instance in Table II are shown in Figs. 3–6. In these Gantt charts, the top number represents the workpiece number, and the bottom number represents the operation number. LF_f means loading the workpiece on F_f, and UF_f means unloading the workpiece from F_f.

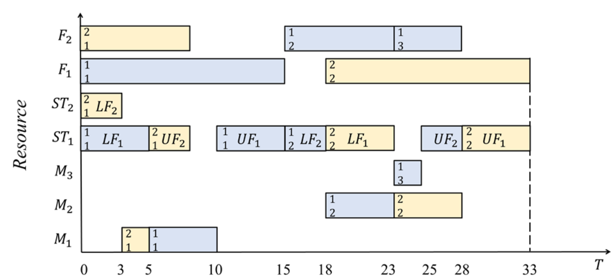


Fig. 6. Gantt chart of MRFJSP with FPs, STs, and LUT.

TABLE IV
NUMBER OF COMBINATIONS FOR RESOURCE SELECTION, OPERATION SEQUENCE, AND SCHEDULE SCHEME

	Machine selection	FP selection	ST selection	Operation sequence	Schedule scheme
Fig. 3	36	-	-	10	360
Fig. 4	36	4	-	10	1440
Fig. 5	36	4	1	10	1440
Fig. 6	36	4	8 or 32	10	11520 or 46080

Fig. 3 shows the common FJSP, where the machine is the only resource and the makespan totally depends on processing time. Fig. 4 illustrates MRFJSP with FPs and LUT, which was studied by Zhou et al. [35]. Figs. 5 and 6 explain that the MRFFS studied in this article, with the main difference being the number of STs. Unlike the problem in Fig. 4, limited STs are considered, meaning loading and unloading need to be queued. FPs are occupied from the start of loading to the end of unloading. In Fig. 6, the completed time goes from 36 to 33 due to O_{11} and O_{21} being loaded at two STs at the same time.

Table IV shows the increasing complexity across the four cases by listing the number of combinations of resource selection, operation sequencing, and the decision space. The growth comes from various resource selection and the different possible orders of loading, processing, and unloading, which must be modeled as three separate segments. At the operation level, these segments can be noncontiguous or absent, and their resource-occupancy windows differ: machine only during processing; ST only during loading/unloading; and FP held continuously from the start of loading to the end of unloading.

C. Mathematical Modeling of MRFFS

Nomenclature shows the notations used in the model. There are $|I|$ workpieces, each of which has $|J_i|$ operations to be processed. O_{ij} is the jth operation of the workpiece i, which has an available machine set M_{ij} with processing time T_{ijm} and an available FP set F_{ij} with loading or unloading time E_f. $|N|$ STs are used to load and unload workpieces.

Assumptions are described as follows.

1) When time = 0, all machines, FPs, and STs are available.
2) Due to the mobility and universality of FPs, they can be utilized by all machines within the PAS, and the combination of fixtures and pallets remains unchanged.
3) Workpieces need to be clamped on FPs both during transportation by the stacker crane and during processing by the machine.
4) The LUT of a workpiece cannot be ignored, and it varies with different FPs.
5) The time of moving FP or FPW by the stacker crane is ignored because the stacker crane operates at high speeds.
6) The processing of the operation is supposed to be continuous, and breakdown is not allowed in principle once started.

The objective function, special constraints, and general constraints presented as follows:

1) Objective Function: The objective function is to minimize the maximum completion time as (1). Equation (2) ensures that the makespan is no less than the completion time of each operation

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

$$C_{ij} \le C_{\max} \quad \forall i \in I\ \forall j \in J_i. \tag{2}$$

2) Special Constraints: Equation (3) guarantees that each operation can only be fixed by one FP at a time. Equation (4) limits that one FP can only be used by one operation at a time. Equation (5) restricts that once a workpiece is loaded onto an FP, it will occupy the FP until unloaded. Other workpieces cannot use the FP during the occupied time

$$\sum_f^F A_{ijf} = 1 \quad \forall i \in I\ \forall j \in J_i \tag{3}$$

$$\begin{aligned}\mathrm{CUF}_{ij} \le\ & \mathrm{SLF}_{i'j'} + L \times (2 - A_{ijf} - A_{i'j'f}) \\ & + L \times (1 - Z_{iji'j'f}) \\ & \forall i \in I\ \forall j \in J_i\ \forall i' \in I\ \forall j' \in J_{i'}\ \forall f \in F\end{aligned} \tag{4}$$

$$\begin{aligned}\mathrm{CUF}_{ij} \le\ & \mathrm{SLF}_{i'j'} + L \times \left(4 - A_{ijf} - A_{i'j'f} - \sum_n^N \mathrm{IUF}_{ijn} \right. \\ & \left. - \sum_n^N \mathrm{ILF}_{i'j'n}\right) \\ & + L \times (1 - UL_{iji'j'f}) \\ & \forall i \in I\ \forall j \in J_i\ \forall i' \in I\ \forall j' \in J_{i'}\ \forall f \in F.\end{aligned} \tag{5}$$

Equations (6) and (7) guarantee that loading/unloading is necessary for the first/final operation of any workpieces. Equations (8) and (9) are explained as follows. If the adjacent operations of the same workpiece use the same FP, no unloading for the preceding operation, and no loading for the subsequent one. Otherwise, the workpiece is loaded and unloaded at one ST

$$\sum_n^N \mathrm{ILF}_{ijn} = 1 \quad \forall i \in I,\ j = 1 \tag{6}$$

$$\sum_n^N \mathrm{IUF}_{ijn} = 1 \quad \forall i \in I,\ j = |J_i| \tag{7}$$

$$\begin{aligned}\sum_n^N \mathrm{ILF}_{ijn} = & \sum_f^F (A_{ijf} \times (1 - A_{i(j-1)f})) \\ & \forall i \in I,\ j = 2, 3, \ldots, |J_i|\end{aligned} \tag{8}$$

$$\begin{aligned}\sum_n^N \mathrm{IUF}_{ijn} = & \sum_f^F (A_{ijf} \times (1 - A_{i(j+1)f})) \\ & \forall i \in I,\ j = 1, 2, \ldots, (|J_i| - 1).\end{aligned} \tag{9}$$

Equations (10)–(13) ensure that one ST can only load or unload one operation at a time

$$\begin{aligned}\mathrm{CUF}_{ij} \le\ & \mathrm{SLF}_{i'j'} + L(2 - \mathrm{IUF}_{ijn} - \mathrm{ILF}_{i'j'n}) \\ & + L(1 - UL_{iji'j'n}) \\ & \forall i \in I\ \forall j \in J_i\ \forall i' \in I\ \forall j' \in J_{i'}\ \forall n \in N\end{aligned} \tag{10}$$

$$\begin{aligned}\mathrm{CLF}_{ij} \le\ & \mathrm{SLF}_{i'j'} + L(2 - \mathrm{ILF}_{ijn} - \mathrm{ILF}_{i'j'n}) \\ & + L(1 - LL_{iji'j'n}) \\ & \forall i \in I\ \forall j \in J_i\ \forall i' \in I\ \forall j' \in J_{i'}\ \forall n \in N\end{aligned} \tag{11}$$

$$\begin{aligned}\mathrm{CUF}_{ij} \le\ & \mathrm{SUF}_{i'j'} + L(2 - \mathrm{IUF}_{ijn} - \mathrm{IUF}_{i'j'n}) \\ & + L(1 - UU_{iji'j'n}) \\ & \forall i \in I\ \forall j \in J_i\ \forall i' \in I\ \forall j' \in J_{i'}\ \forall n \in N\end{aligned} \tag{12}$$

$$\begin{aligned}\mathrm{CLF}_{ij} \le\ & \mathrm{SUF}_{i'j'} + L(2 - \mathrm{ILF}_{ijn} - \mathrm{IUF}_{i'j'n}) \\ & + L(1 - LU_{iji'j'n}) \\ & \forall i \in I\ \forall j \in J_i\ \forall i' \in I\ \forall j' \in J_{i'}\ \forall n \in N.\end{aligned} \tag{13}$$

3) General Constraints: Equation (14) limits that each operation can only be processed on one machine at a time. Equation (15) guarantees that a machine can only process one operation at a time

$$\sum_m^M X_{ijm} = 1 \quad \forall i \in I\ \forall j \in J_i \tag{14}$$

$$\begin{aligned}C_{ij} \le\ & S_{i'j'} + L(2 - X_{ijm} - X_{i'j'm}) + L(1 - Y_{iji'j'm}) \\ & \forall i \in I\ \forall j \in J_i\ \forall i' \in I\ \forall j' \in J_{i'}\ \forall m \in M.\end{aligned} \tag{15}$$

Equations (16)–(21) restrict that the loading, processing, and unloading of the workpiece are carried out in sequence

$$\mathrm{CLF}_{ij} \le S_{ij} \quad \forall i \in I,\ j = 1, 2, \ldots, |J_i| \tag{16}$$

$$C_{ij} \le \mathrm{SUF}_{ij} \quad \forall i \in I,\ j = 1, 2, \ldots, |J_i| \tag{17}$$

$$\mathrm{CUF}_{i(j-1)} \le \mathrm{SLF}_{ij} \quad \forall i \in I,\ j = 2, \ldots, |J_i| \tag{18}$$

$$\mathrm{SLF}_{ij} + \sum_n^N \mathrm{ILF}_{ijn} \times \sum_f^F (A_{ijf} \times E_f) \le \mathrm{CLF}_{ij}$$

$$\forall i \in I,\ \ j = 1, 2, \ldots, |J_i| \tag{19}$$

$$\text{SUF}_{ij} + \sum_{n}^{N} \text{IUF}_{ijn} \times \sum_{f}^{F} (A_{ijf} \times E_f) \leq \text{CUF}_{ij}$$

$$\forall i \in I,\ \ j = 1, 2, , \ldots, |J_i| \tag{20}$$

$$S_{ij} + \sum_{m}^{M} (T_{ijm} \times X_{ijm}) \leq C_{ij} \quad \forall i \in I \quad \forall j \in J_i. \tag{21}$$

III. Search Algorithm Based on Critical Paths and Points

To solve the complicated problem proposed in this study, SACP is developed. The four main steps are described as follows.

Step 1 (Initialization):

The four-layer encoding scheme is applied. MS and FS are initialized based on rules. SS is generated based on FS, and OS is grouped and disrupted based on FS.

Step 2 (Decoding):

A new decoding method, TPI-IAR, has been developed, which has two stages: reading time periods and finding available time intervals.

Step 3 (Outer Loop: Crossover and Ranking Selection):

Three different crossover operators are selected based on a fixed proportion. Ranking selection is applied to choose a set number of better solutions to form a new population of individuals.

Step 4 (Inner Loop: Mutation and Feasibility Correction):

Three different mutation operators to MS and FS based on critical paths and points are chosen according to self-learning weights. Feasibility correction can regenerate the SS and OS to ensure feasible solutions before decoding.

Moreover, the detailed flowchart of SACP is illustrated in Fig. 7. After encoding and initialization, the individuals are decoded with TPI-IAR, while the best x solutions are added to the *Elite Pool*. Termination condition *I* indicates that if the number of iterations reaches L or the global optimal solution does not improve for y successive generations, the loop ends and the current global optimal solution is output. Otherwise, the loops are repeated, and the *Elite Pool* is updated. The crossover rate (pc) determines how many individuals in the current population need to select a crossover operator from three options. According to the mutation rate (pm), individuals are mutated by one of the three operators based on critical paths and points with self-learning weights. After mutation, SS and OS are repaired to ensure feasibility, and the new individuals are decoded. If the mutated solution outperforms the original by a factor of the acceptable rate, the operator's score increases. Termination condition II means that if the result is within the acceptable range, the mutation of the current individual is completed. Otherwise, the current individual continues to mutate until the number of mutations reaches N. After all necessary mutations have been completed, the new generation is generated by ranking selection.

A. Step 1: Initialization

The encoding structure is described in Fig. 8. MS, FS, and SS are all arranged in the sequence of operations from O_{11} to O_{22}. The number in each gene of MS/FS represents the index of the selected machine/FP from the corresponding operation's alternative machine/FP set. In SS, one operation corresponds to two genes, where the first is the loading gene and the second is the unloading gene. The number within OS signifies

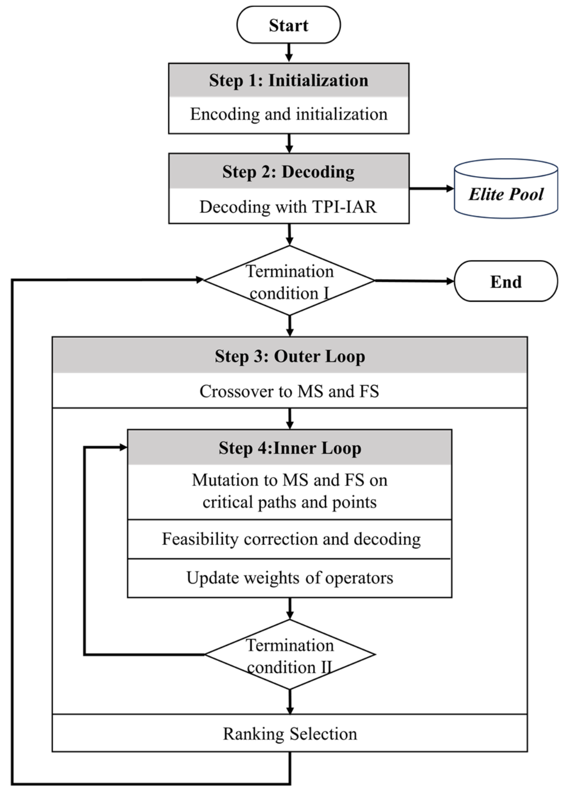


Fig. 7. Flowchart of SACP.

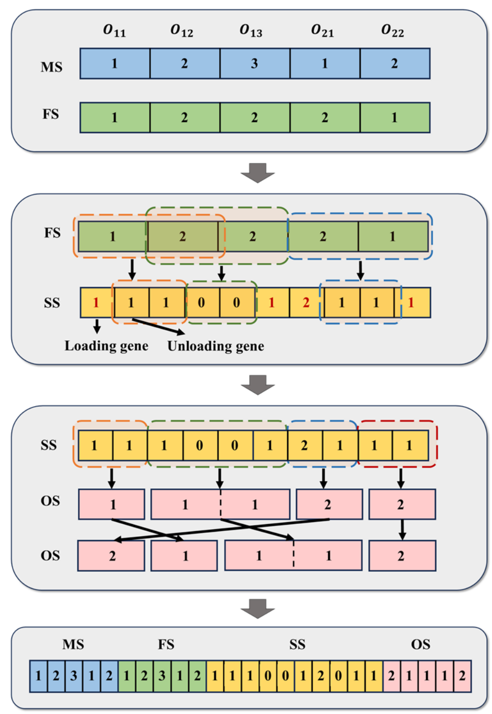


Fig. 8. Four-layer encoding scheme.

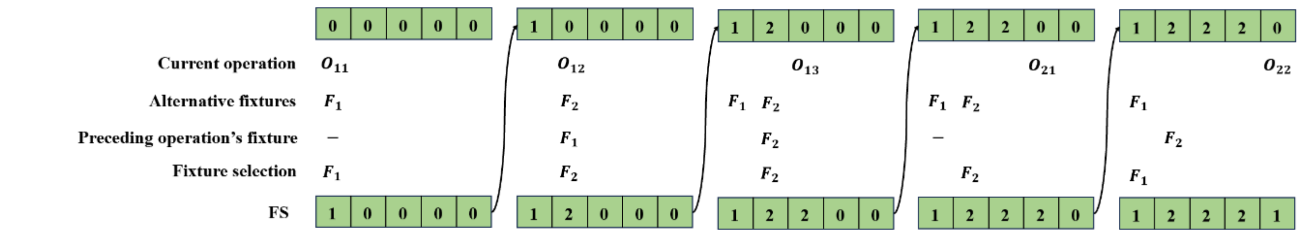


Fig. 9. Minimum number of LUT rules.

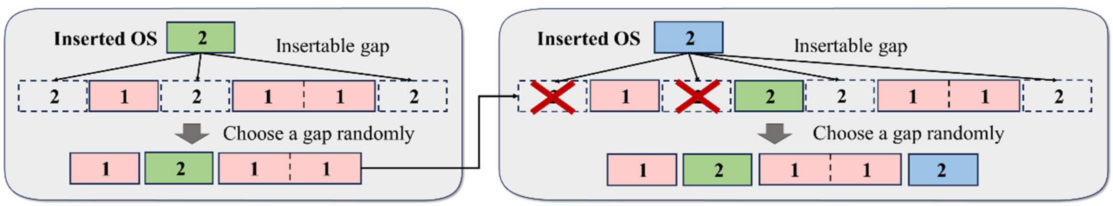


Fig. 10. Explanation for Rule 7.

the workpiece index, with its sequence indicating the specific operation associated with the gene. Thus, the initial OS is [1, 1, 1, 2, 2].

The initialization of MS and FS can be random or based on predefined rules. Two rules are specified as follows.

Rule 1 (Minimum Machine Load Rule (MML) [51]): Select the machine with the lowest accumulated processing time from the operation's alternative machine set for each operation.

Rule 2 [Minimum Number of LUT Rule (MNLU)]: For adjacent operations of the same workpiece, if the FP selected by the previous operation is in the alternative FP set of the current operation, the same FP is chosen to reduce the number of loading and unloading. For the first operation of each workpiece, if more than one FP is available, select it randomly (see Fig. 9).

In each gene of SS, nonzero numbers indicate the index of ST, and zero means no loading or unloading. For example, the first and the second numbers "1" mean that the operation O_{11} needs to be loaded at ST_1 and unloaded also at ST_1. Three significant rules need to be followed during SS initialization.

Rule 3: The first operation of each workpiece needs to be loaded, and the final operation needs to be unloaded.

Rule 4: According to FS, if the adjacent operations of each workpiece use different FPs, the previous operation needs to be unloaded, and the current operation needs to be loaded. Otherwise, there is no loading or unloading between these two operations.

Rule 5: If loading or unloading is necessary, the ST is randomly selected from all the alternatives in PAS.OS is first grouped according to Rule 6, which means operations of each workpiece using the same FP have the same priority. This rule ensures that any two consecutive operations that require no loading/unloading are scheduled adjacent. Then, the order between grouped operations is disrupted based on Rule 7, but the order within them is maintained. Rule 7 ensures that an FP will not be used by another workpiece when occupied by the current workpiece. As shown in Fig. 10, the first operation of workpiece 2 has three insertable gaps, and a random choice leads to the sequence [1], [2], and [1, 1]. For the second operation of workpiece 2, only two gaps are available since it has to be inserted after the first operation, rendering the first two gaps unusable.

Rule 6: OS is grouped according to SS from left to right to ensure that there are two nonzero numbers in each group.

Rule 7: The first workpiece determines the initial sequence, with subsequent operations inserted into insertable gaps in order. Ensure that for a given workpiece (except for the first), the following operations are only inserted into a gap after the previous operation of the same workpiece, maintaining the order within the groups.

TABLE V
STAGE 1 OF TPI-IAR

Stage 1: Decode MS, FS, and SS in OS order and get all time periods of processing, loading, and unloading

input : OS, MS, FS, SS
output: m(selected machine), f(selected FP), n_1(loading ST), n_2(unloading ST), processing time, loading time, unloading time for O_{ij}(each operation)

Assume that the total operation number of all workpieces is S;
for $s = 1$ **to** S **do**
  $i \leftarrow$ the value at the s-th positison in the OS;
  $j \leftarrow$ the number of i so far;
  $m \leftarrow$ the value at the $(\sum_{n=0}^{i-1} J_n + j)$-th position in the MS;
  $f \leftarrow$ the value at the $(\sum_{n=0}^{i-1} J_n + j)$-th position in the FS;
  $n_1 \leftarrow$ the value at the $(2(\sum_{n=0}^{i-1} J_n + j) - 1)$-th position in the SS;
  $n_2 \leftarrow$ the value at the $(2(\sum_{n=0}^{i-1} J_n + j))$-th position in the SS;
  processing time $\leftarrow \mathrm{T}_{ijm}$ **if** $n_1 = 0$ **then** $loadingtime = 0$;
  else loading time=E_f;
  if $n_2 = 0$ **then** unloading time=0;
  else unloading time=E_f;

B. Step 2: Decoding

TPI-IAR consists of two stages. The key to the first stage (Table V) is to decode the OS part into the scheduling for the MS, FS, and SS, and to read the processing time, loading time, and unloading time of each operation. The key to the second stage (Table VI) is to identify the time intervals when all the required resources are available and arrange the start

TABLE VI
STAGE 2 OF TPI-IAR

Stage 2: Identify the time intervals when all the required resources are available and determine whether the time period can be inserted

input : m, f, n_1, n_2, processing time, loading time, unloading time for O_{ij}
output: S_{ij}(start processing time), C_{ij}(end processing time), SLF_{ij}(start loading time), CLF_{ij}(end loading time), SUF_{ij}(start unloading time), CUF_{ij}(end unloading time) for O_{ij}

Assume that the total operation number of all workpieces is S;
for $s = 1$ **to** S **do**
  Step1: arrange SLF_{ij} and CLF_{ij} for O_{ij};
  Step2: arrange S_{ij} and C_{ij} for O_{ij};
  Step3: arrange SUF_{ij} and CUF_{ij} for O_{ij};

TABLE VII
STEP 1 OF STAGE 2

Step 1: Arrange SLF_{ij} and CLF_{ij} for O_{ij}

if n_1=*0* **then** Find all available intervals for FP f, and the number is X
  for $x = 1$ **to** X **do**
    SLF_{ij}=max{the end unloading time of the previous operation $O_i(j-1)$, the start time of the interval x};
    CLF_{ij}=SLF_{ij} + loading time;
    if CLF_{ij} <= *the end time of interval* x **then** Record SLF_{ij} and CLF_{ij}
if n_1=*0* **then** Find all available intervals for FP f and ST n_1 , then find the intersections of two group intervals, and the number of intersections is Y;Find the end unloading time of FP f used for the previous operation that needs to be unloaded
  for $y - 1$ **to** Y **do**
    SLF_{ij}=max{the end unloading time of the operation $O_{i(j-1)}$, the start time of the intersection y, the previous end unloading time of FP f};
    CLF_{ij}=SLF_{ij} + loading time;
    if CLF_{ij} <= *the end time of interval* y **then** Record SLF_{ij} and CLF_{ij}

TABLE VIII
STEP 2 OF STAGE 2

Step 2: Arrange S_{ij} and C_{ij} for O_{ij}

Find all available intervals for machine m and FP f ,then find the intersections of two group intervals, and the number of intersections is Z
for $z = 1$ **to** Z **do**
  S_{ij}=max{the end loading time of the operation O_{ij} , the start time of the interval z};
  C_{ij}=S_{ij} + processing time;
  if C_{ij} <= *the end time of interval* z **then** Record S_{ij} and C_{ij}

and end times for the processing, loading, and unloading activities. The pseudocodes of the loading time decision and processing time decision are shown in Tables VII and VIII. The unloading decision is similar to the loading decision, so the code explanation will not be repeated.

C. Outer Loop: Crossover and Ranking Selection

Three various crossover operators for MS and FS of the individuals that need to cross are explained as follows.

Operator 1: Crossing with one of the individuals from the current population.

Operator 2: Crossing with one of the individuals from the Elite Pool.

Rule 8 [Uniform Crossover of Random Number (UCRN) of Genes]: Several randomly selected genes in the FS or MS length range are stored in set S_1, and the remaining unselected genes are stored in set S_2. A new individual C_1 is formed by preserving the gene of the first parent belonging to S_1 and the gene of the second parent belonging to S_2. Another new individual C_2 is created at the same time.

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

D. Inner Loop: Mutation and Feasibility Correction

1) Critical Paths and Points: Critical paths include processing, loading, and unloading periods. The explanation is depicted in Fig. 11, as illustrated by the following two definitions.

Definition 1 (Critical Path): Starting with a time period whose start time equals zero, find a path such that the end time of the previous time period coincides with the start time of the subsequent time period, meaning there is no idle time in between, and finally, end the path with a time period that ends at the makespan.

Definition 2 (Critical Points): On the critical path, if more than one time period starts at the end time of the current time period, then the current time period is a critical point.

2) Mutation Operators: Three various mutation operators for MS and FS of the individuals are listed as follows.

Operator 1: Reselecting the machine or FP for all critical points on the critical path.

Operator 2: Reselecting the machine or FP for all noncritical points on the critical path.

Operator 3: Reselecting the machine or FP for any four genes.

3) Feasibility Correction: From the encoding strategy, the SS completely depends on FS, and the OS depends on SS to be grouped and disrupted. Therefore, if the FS changes but the SS and OS do not change, encoding errors will be caused. Feasibility correction can regenerate the SS and OS based on Rules 3–7 before decoding.

IV. CASE STUDY

A. Design of Experiments

All experiments were conducted in a desktop computer with an Intel[1] Core[2] i5-10210U CPU @ 1.60 2.11 GHz, Windows 10, PyCharm 2023.2.4 (Professional Edition), and Python 3.11.5.

The study focuses on a PAS from one of the largest Chinese engine manufacturing enterprises, as shown in Fig. 12. The 15 workpiece, machine, FP, and ST (WMFS) instances were generated from real data and divided into three different scales, which, respectively, correspond to 0–100, 100–200, and above 200 operations, as listed in Table IX.

The parameters of SACP are set as summarized in Table X, which was determined based on a series of experiments. The

[1]Registed trademark.
[2]Trademarked.

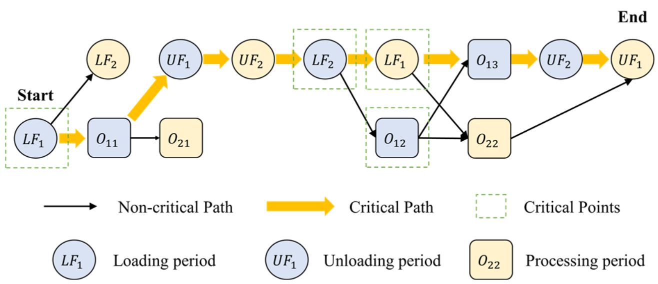


Fig. 11. Critical path and points based on Fig. 5.

Fig. 12. PAS in the engine manufacturing enterprise.

TABLE IX
THREE SCALES OF WMFS INSTANCES

Scale	WMFS	Workpiece	Operation	Machine	FP	ST
Small	1	3	3	2	2	1
	2	3	3	3	2	1
	3	5	3	5	5	1
	4	8	6	6	5	1
	5	10	5	5	6	1
Medium	6	10	10	8	8	2
	7	10	12	8	8	2
	8	12	15	10	10	3
	9	12	15	12	10	3
	10	15	10	12	12	3
Large	11	20	10	12	15	4
	12	20	14	12	15	4
	13	30	15	15	18	5
	14	40	15	15	18	5
	15	50	9	18	20	5

subsequent numerical experiments are structured to assess the efficacy of SACP designed for addressing MRFFS and to offer practical ST allocation strategies for industrial applications.

B. Performance of SACP

To verify the accuracy of the mathematical model and the performance of the proposed method, SACP is compared with GUROBI, genetic algorithm (GA), simulated annealing (SA), tabu search (TS), and the adaptive large neighborhood search algorithm (ALNS) on 15 WMFS.

GUROBI is a high-performance mathematical optimization solver, widely used in linear programming, mixed-integer programming, quadratic programming, and other optimization problems. It approximates the global optimal solution through methods such as branch-and-bound, cutting planes, or heuristic algorithms. Clear definitions of variables and constraints are essential; otherwise, GUROBI may fail to converge and obtain a feasible solution. GUROBI can find the optimal solution, a near-optimal solution, or fail to find any solution within the

TABLE X
PARAMETERS FOR SACP

Notation	Meaning	Value
m	Population scale in the algorithm	100
L	The limitation of the number of iterations of the outer loop	150
N	The limitation of the number of iterations of inner loop	5
y	The iteration stops when the global optimal value is unchanged for limited consecutive generations	60
x	The number of elite solutions in the Elite Pool	20
GL, RL	The proportion of initialization based on MML, MLUT and random.	0.6, 0.4
pc, pm	The proportion of crossover and mutation	0.7,0.3
$p1$, $p2$, $p2$	The proportion of three crossover operators	0.6, 0.2, 0.2
$w1$, $w2$, $w3$	The initial weights of three mutation operators	1.0, 1.0, 1.0
a	Acceptable range means allowing inferior solution within a certain range	1.2
b	The awarded score of the mutation operator while it obtains the solution within the acceptable range	0.01

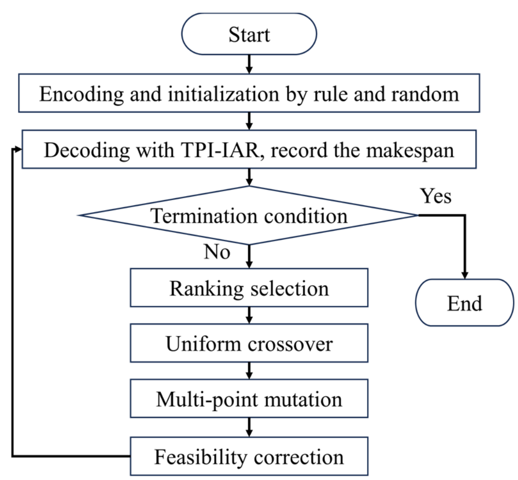


Fig. 13. Flowchart of GA.

specified time. The time limits of the GUROBI mathematical programming solver for three scales are set to 3600, 7200, and 14 400 s.

In GA, the population size, iterations, pc, pm, initialization, and the limited consecutive generations are the same as those in Table X. In SA, the iterations L are set to 10 000, the limited consecutive generations are 6000, the initial temperature is 200, and N is 10. In TS, the iterations L is set to 10 000, and the number of rows in the tabu table is limited to 5. In ALNS, the iterations L are set to 1000, the initial temperature is 200, and N is 10. The flowcharts of GA, SA, TS, and ALNS are shown in Figs. 13–16.

TABLE XI
RESULTS OF COMPARISON BETWEEN SIX METHODS AND 15 INSTANCES

WMFS	GUROBI	GA	SA	TS	ALNS	SACP	Q_1	Q_2	Q_3	Q_4	Q_5
	$\overline{Opt_1}$	$\overline{Opt_2}$	$\overline{Opt_3}$	$\overline{Opt_4}$	$\overline{Opt_5}$	$\overline{Opt_6}$					
1	57	57.0	57.0	57.0	57.0	57.0	0.00%	0.00%	0.00%	0.00%	0.00%
2	76	76.0	76.0	76.0	76.0	76.0	0.00%	0.00%	0.00%	0.00%	0.00%
3	112	115.1	154.4	123.1	114.5	114.0	1.79%	-0.96%	-26.17%	-7.39%	-0.44%
4	-	283.1	400.9	307.1	278.9	269.9	-	-4.66%	-32.68%	-12.11%	-3.23%
5	-	771.0	1044.1	800.5	768.8	738.4	-	-4.23%	-29.28%	-7.76%	-3.95%
6	-	412.9	442.3	403.3	386.7	317.7	-	-23.06%	-28.17%	-21.22%	-17.84%
7	-	516.8	597.4	579.0	507.9	486.7	-	-5.82%	-18.53%	-15.94%	-4.17%
8	-	818.3	843.8	837.8	789.7	739.1	-	-9.68%	-12.41%	-11.78%	-6.41%
9	-	810.3	827.5	823.0	798.6	751.0	-	-7.32%	-9.24%	-8.75%	-5.96%
10	-	537.9	546.5	523.2	506.7	487.6	-	-9.35%	-10.78%	-6.80%	-3.77%
11	-	582.8	577.2	568.0	568.8	537.0	-	-7.86%	-6.96%	-5.46%	-5.59%
12	-	1170.8	1045.2	1109.4	1089.0	1004.2	-	-14.23%	-3.92%	-9.48%	-7.79%
13	-	1511.0	1453.0	1469.4	1479.5	1409.8	-	-6.70%	-2.97%	-4.06%	-4.71%
14	-	1803.2	1790.3	1785.3	1780.0	1731.4	-	-3.98%	-3.29%	-3.02%	-2.73%
15	-	931.2	953.7	935.0	970.2	914.6	-	-1.78%	-4.10%	-2.18%	-5.73%

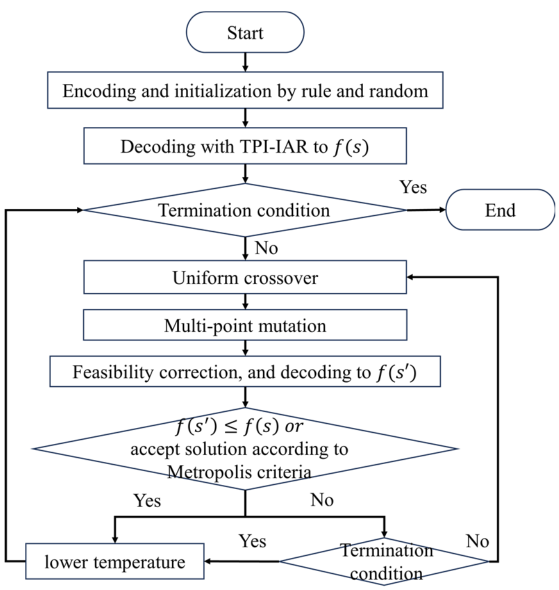


Fig. 14. Flowchart of SA.

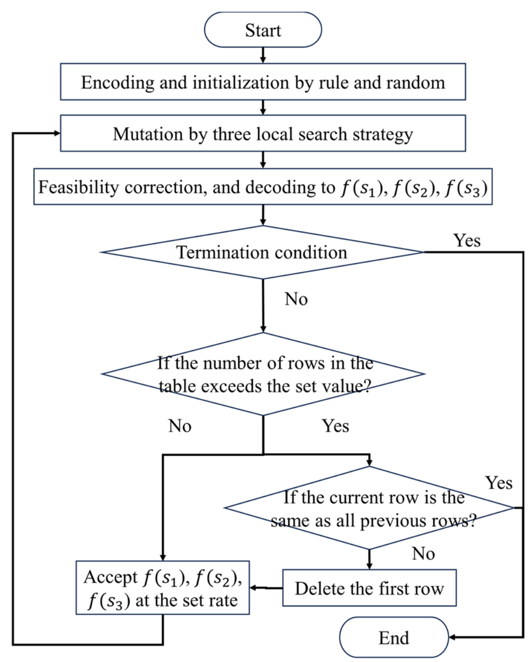


Fig. 15. Flowchart of TS.

The metric Q_i is used to measure the performance of the algorithms, which is defined by (22). Here, $\overline{\text{Opt}_1}$ represents the optimal solution by GUROBI within the limited time, while $\overline{\text{Opt}_2}$, $\overline{\text{Opt}_3}$, $\overline{\text{Opt}_4}$, $\overline{\text{Opt}_5}$, and $\overline{\text{Opt}_6}$ indicate the average values of ten times of GA, SA, TS, ALNS, and SACP

$$Q_i = \frac{\overline{\text{Opt}_6} - \overline{\text{Opt}_i}}{\overline{\text{Opt}_i}} \times 100\%, \quad i = 1, 2, 3, 4, 5. \tag{22}$$

Table XI compares the experimental outcomes for solving WMFS, revealing the following specific insights.

Insight 1.1: Most of the metrics Q_i are negative, indicating that the proposed algorithm SACP generally outperforms GUROBI, GA, SA, TS, and ALNS.

Insight 1.2: For WMFS1 and WMFS2, GUROBI, GA, SA, TS, ALNS, and SACP can all obtain the same optimal solution, validating the model, encoding, and decoding's accuracy.

Insight 1.3: The complexity of the problem increases with the number of various resources, preventing GUROBI from obtaining a feasible solution in the set time for WFMS4–WFMS15. Therefore, a meta-heuristic algorithm is necessary.

C. Performance of Critical Paths and Points Mutation

By comparing SACP with SACP-CP (SACP without the critical paths and points mutation; all other components unchanged) on the 15 WMFS instances, the effectiveness of

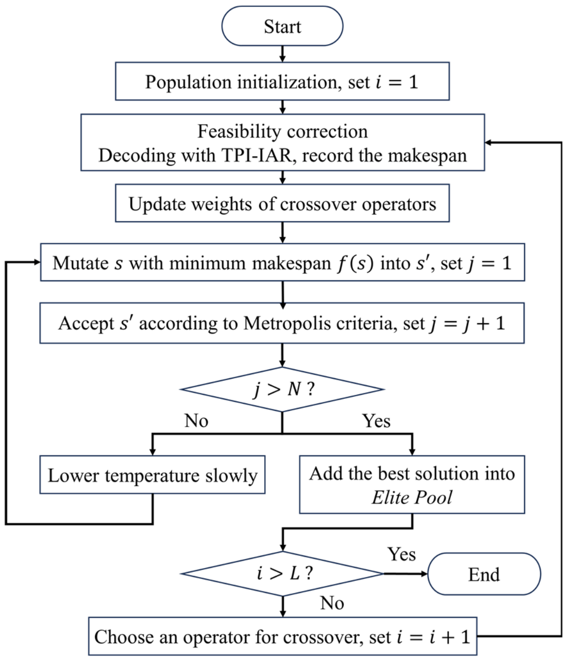


Fig. 16. Flowchart of ALNS.

TABLE XII
PERFORMANCE COMPARISON OF SACP AND SACP-CP ON WMFS

WMFS	SACP	SACP-CP
1	57.0	57.0
2	76.0	76.0
3	114.0	115.2
4	269.9	284.6
5	738.4	774.6
6	317.7	333.6
7	486.7	511.8
8	739.1	811.1
9	751.0	816.9
10	487.6	544.8
11	537.0	585.2
12	1004.2	1087.5
13	1409.8	1494.1
14	1731.4	1804.7
15	914.6	933.7

the mutation method is verified. For each instance and each method, ten independent runs under the same initialization are performed. Table XII summarizes the results of the average makespan over ten runs. The conclusions are listed as follows.

Insight 2.1: SACP achieves a lower average makespan than SACP-CP on the majority of WMFS instances (with ties on a few cases). This indicates that the critical paths and points mutation materially improve search quality by guiding perturbations to bottleneck regions, thereby enhancing solution effectiveness.

D. Improvements in Real Orders

In this study, we focus on largest Chinese engine manufacturing enterprises that currently employ manual scheduling for production planning. In actual production, workshop orders typically involve multiple products, with complex production processes and high demands for resource allocation. To demonstrate the practical value of the proposed SACP, we conducted a comparative analysis using two representative actual orders, containing five and ten different products, respectively. Table XIII presents a comparison between manual scheduling and SACP in terms of makespan, average machine utilization (U_1), average FP utilization (U_2), and average ST utilization (U_3). The calculation method for average resource utilization (U_n) is shown in (23), where N_1, N_2, and N_3, respectively, represent the number of machines, FPs, and STs. T_i indicates the total occupied time of the resource i

$$U_n = \frac{\sum_{i=1}^{N_n} T_i}{N_n \times \text{makespan}}, \quad (n = 1, 2, 3). \tag{23}$$

The insights are listed as follows.

Insight 3.1: For the order with five products, the SACP algorithm reduced the makespan by 38.30% and improved the utilization rates of machine, FP, and ST resources by 72.07%, 38.53%, and 3.95%, respectively.

Insight 3.2: For the order with ten products, the SACP algorithm shortened the makespan by 23.08% and increased the utilization rates of machine, FP, and ST resources by 29.26%, 35.47%, and 14.15%, respectively.

Insight 3.3: SACP significantly improves production efficiency and optimizes resource allocation. It provides an efficient and reliable scheduling solution for the company's actual production.

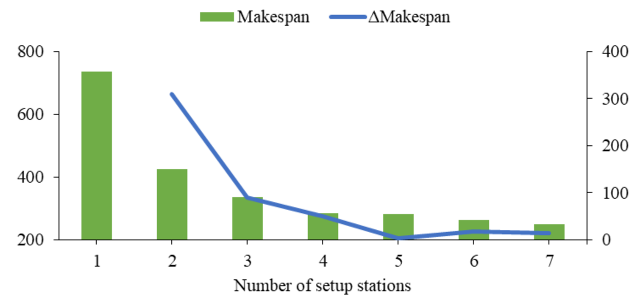


Fig. 17. Graphical result of WMFS5.

E. Setup Station Allocation Experiments

In the preliminary resource allocation of actual production, determining the appropriate number of STs for a PAS is critical. Enterprises are most concerned about enhancing production efficiency by strategically adjusting the number of STs without additional resource investment when deciding the allocation of STs.

This experiment aims to explore the impact of the number of STs on the makespan and the reduction in makespan between consecutive ST levels (ΔMakespan). With all other parameters held constant and only the number of STs varied, experiments were conducted on WMFS5 and WMFS10, ranging from 1 to 7 STs. Each experiment was tested five times. Figs. 17 and 18 indicate that increasing the number of STs correlates with reducing makespan, with insights as follows.

TABLE XIII
COMPARISON OF THE RESULTS OF MANUAL SCHEDULING AND IGARC

Order	Method	Makespan	U_1	U_2	U_3
Order with five products	Manual scheduling	141	25.25%	47.63%	86.25%
	SACP	87	43.45%	65.98%	89.66%
Order with ten products	Manual scheduling	377	37.13%	70.66%	43.50%
	SACP	290	48.00%	95.73%	49.66%

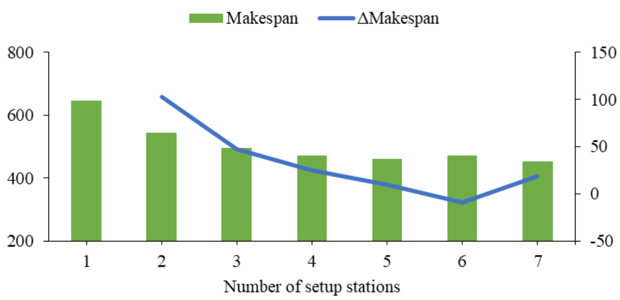


Fig. 18. Graphical result of WMFS10.

Insight 4.1: A significant makespan decrease occurs when the number of STs increases from 1 to 2, with a minor reduction as more STs are added.

Insight 4.2: With a reasonable allocation of STs, enterprises can shorten the makespan, alleviate congestion at one ST, and maintain a balanced utilization rate, thus avoiding unnecessary resource waste.

Therefore, for enterprises, increasing the number of STs from 1 to 2 is an effective strategy for improving the production efficiency of PAS. However, if considering further expanding the number of STs, it is necessary to evaluate the relationship between additional costs and potential benefits to ensure the rational allocation of resources and the maximization of cost-effectiveness.

V. CONCLUSION

In this article, a new multiresource-constrained flexible PAS scheduling problem that accounts for machines, FPs, STs, and LUTs is studied. We formulated the problem as a mixed-integer programming model and proposed a new search algorithm based on critical path and point (SACP) with a new decoding method, TPI-IAR.

Overall, the results validate the proposed model and demonstrate the effectiveness of SACP. Specifically, the following conditions hold.

1) *Benchmark Performance and Validation:* SACP outperforms GUROBI and GA/SA/TS/ALNS on most instances, and matches the optimal solutions on the smallest cases, confirming the model and encoding/decoding correctness.
2) *Effectiveness of the Critical Path Search Mechanism:* The ablation results show that the critical path and point mutation improve search quality and reduce the average makespan compared with the variant without this operator.
3) *Practical Value in Real Production:* On real orders, SACP reduces makespan and improves machine/FP/ST utilization.
4) *Setup Station Allocation Value:* Increasing STs from 1 to 2 brings the largest makespan reduction and helps relieve congestion and balance resource usage.

This study has several limitations. First, the problem is studied in a deterministic and static setting, where order arrivals, processing/setup times, and resource availability are assumed to be known in advance and do not change during execution. Second, we focus on minimizing makespan, while other performance measures that are often important in practice are not explicitly considered. Third, the proposed approach is developed for offline planning; when disruptions occur, fast rescheduling and real-time decision support are still needed.

Therefore, future work will focus on: 1) extending the model to dynamic and uncertain environments (e.g., stochastic arrivals, machine/FP/ST unavailability, and order insertions); 2) incorporating additional practical objectives such as tardiness and energy-related costs; and 3) developing fast online rescheduling strategies. In particular, reinforcement learning is a promising direction to learn dispatching/rescheduling policies offline and enable rapid decisions when disruptions occur.

REFERENCES

[1] F. T. S. Chan and H. K. Chan, "A comprehensive survey and future trend of simulation study on FMS scheduling," *J. Intell. Manuf.*, vol. 15, no. 1, pp. 87–102, Feb. 2003, doi: 10.1023/b:jims.0000010077.27141.be.

[2] B. L. Maccarthy and J. Liu, "A new classification scheme for flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 31, no. 2, pp. 299–309, Feb. 1993, doi: 10.1080/00207549308956726.

[3] Y. Cheng, Z. Xie, Y. Xin, K. Chen, and R. Zarei, "Flexible job shop scheduling method for optimizing mold resource setup time," *IEEE Access*, vol. 12, pp. 33486–33503, 2024, doi: 10.1109/ACCESS.2024.3372396.

[4] J.-Q. Li, Z.-M. Liu, C. Li, and Z.-X. Zheng, "Improved artificial immune system algorithm for type-2 fuzzy flexible job shop scheduling problem," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 11, pp. 3234–3248, Nov. 2021, doi: 10.1109/TFUZZ.2020.3016225.

[5] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1430–1442, Mar. 2021, doi: 10.1109/TCYB.2019.2939219.

[6] C. Lin, Z. Cao, and M. Zhou, "Learning-based grey wolf optimizer for stochastic flexible job shop scheduling," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3659–3671, Oct. 2022, doi: 10.1109/TASE.2021.3129439.

[7] G. A. Kasapidis, D. C. Paraskevopoulos, I. Mourtos, and P. P. Repoussis, "A unified solution framework for flexible job shop scheduling problems with multiple resource constraints," *Eur. J. Oper. Res.*, vol. 320, no. 3, pp. 479–495, Feb. 2025, doi: 10.1016/j.ejor.2024.08.010.

[8] Q. Perrachon, A.-L. Olteanu, M. Sevaux, S. Fréchengues, and J.-F. Kerviche, "Industrial multi-resource flexible job shop scheduling with partially necessary resources," *Eur. J. Oper. Res.*, vol. 320, no. 2, pp. 309–327, Jan. 2025, doi: 10.1016/j.ejor.2024.07.023.

[9] L. Gao and Q.-K. Pan, "A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem," *Inf. Sci.*, vol. 372, pp. 655–676, Dec. 2016, doi: 10.1016/j.ins.2016.08.046.

[10] X. Liao, R. Zhang, Y. Chen, and S. Song, "A new artificial bee colony algorithm for the flexible job shop scheduling problem with extra resource constraints in numeric control centers," *Expert Syst. Appl.*, vol. 249, Sep. 2024, Art. no. 123556.

[11] Y.-D. Zhou, J.-H. Shin, and D.-H. Lee, "Loading and scheduling for flexible manufacturing systems with controllable processing times," *Eng. Optim.*, vol. 51, no. 3, pp. 412–426, Mar. 2019, doi: 10.1080/0305215x.2018.1469134.

[12] Z. Peng, H. Zhang, H. Tang, Y. Feng, and W. Yin, "Research on flexible job-shop scheduling problem in green sustainable manufacturing based on learning effect," *J. Intell. Manuf.*, vol. 33, no. 6, pp. 1725–1746, Aug. 2022, doi: 10.1007/s10845-020-01713-8.

[13] P. Soofi, M. Yazdani, M. Amiri, and M. A. Adibi, "Robust fuzzy-stochastic programming model and meta-heuristic algorithms for dual-resource constrained flexible job-shop scheduling problem under machine breakdown," *IEEE Access*, vol. 9, pp. 155740–155762, 2021, doi: 10.1109/ACCESS.2021.3126820.

[14] C. Wang, D. Fan, Y. Liu, S. Ren, and J. Wang, "Dual-resource flexible job shop scheduling considering worker proficiency differences," *Comput. Oper. Res.*, vol. 184, Dec. 2025, Art. no. 107216, doi: 10.1016/j.cor.2025.107216.

[15] J. Li, X. Li, L. Gao, C. Wang, and H. Chen, "A multitasking workforce-constrained flexible job shop scheduling problem: An application from a real-world workshop," *J. Manuf. Syst.*, vol. 83, pp. 196–215, Dec. 2025, doi: 10.1016/j.jmsy.2025.09.004.

[16] J. Ahn and H.-J. Kim, "A branch and bound algorithm for scheduling of flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 3, pp. 4382–4396, Jul. 2024, doi: 10.1109/TASE.2023.3296087.

[17] X. Zhu, J. Xu, J. Ge, Y. Wang, and Z. Xie, "Multi-task multi-agent reinforcement learning for real-time scheduling of a dual-resource flexible job shop with robots," *Processes*, vol. 11, no. 1, p. 267, Jan. 2023, doi: 10.3390/pr11010267.

[18] X. Wu, Q. Yuan, and L. Wang, "Multiobjective differential evolution algorithm for solving robotic cell scheduling problem with batch-processing machines," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 757–775, Apr. 2021, doi: 10.1109/TASE.2020.2969469.

[19] 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, Apr. 2023, doi: 10.3390/s23083815.

[20] H. Niu, W. Wu, Z. Xing, X. Wang, and T. Zhang, "A novel multi-tasks chain scheduling algorithm based on capacity prediction to solve AGV dispatching problem in an intelligent manufacturing system," *J. Manuf. Syst.*, vol. 68, pp. 130–144, Jun. 2023, doi: 10.1016/j.jmsy.2023.03.007.

[21] X. Wang, W. Wu, Z. Xing, X. Chen, T. Zhang, and H. Niu, "A neural network based multi-state scheduling algorithm for multi-AGV system in FMS," *J. Manuf. Syst.*, vol. 64, pp. 344–355, Jul. 2022, doi: 10.1016/j.jmsy.2022.06.017.

[22] L. Meng, W. Cheng, C. Zhang, K. Gao, B. Zhang, and Y. Ren, "Novel CP models and CP-assisted meta-heuristic algorithm for flexible job shop scheduling benchmark problem with multi-AGV," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 11, pp. 8455–8468, Nov. 2025, doi: 10.1109/TSMC.2025.3604355.

[23] Q. Liao, J. Yang, and Y. Zhou, "Sustainable scheduling of an automatic pallet changer system by multi-objective evolutionary algorithm with first piece inspection," *Sustainability*, vol. 11, no. 5, p. 1498, Mar. 2019, doi: 10.3390/su11051498.

[24] P. Yunusoglu and S. Topaloglu Yildiz, "Solving the flexible job shop scheduling and lot streaming problem with setup and transport resource constraints," *Int. J. Syst. Sci., Oper. Logistics*, vol. 10, no. 1, Dec. 2023, Art. no. 2221072, doi: 10.1080/23302674.2023.2221072.

[25] J. C. Chen, T.-L. Chen, Y.-Y. Chen, and M.-Y. Chung, "Multi-resource constrained scheduling considering process plan flexibility and lot streaming for the CNC machining industry," *Flexible Services Manuf. J.*, vol. 36, no. 3, pp. 946–993, Sep. 2024, doi: 10.1007/s10696-023-09514-w.

[26] G. Xu, X. Jiang, D. Sun, B. Yang, R. Deng, and J. Fu, "Flexible job-shop scheduling based on improved firefly algorithm," in *Proc. 2nd Int. Conf. Comput.*, Mar. 2022, pp. 90–97, doi: 10.1109/ICCCR54399.2022.9790158.

[27] X. Wu, P. Junjian, X. Xiao, and S. Wu, "An effective approach for the dual-resource flexible job shop scheduling problem considering loading and unloading," *J. Intell. Manuf.*, vol. 32, no. 3, pp. 707–728, Mar. 2020, doi: 10.1007/s10845-020-01697-5.

[28] J. Li, X. Li, L. Gao, and Q. Liu, "A flexible job shop scheduling problem considering on-site machining fixtures: A case study from customized manufacturing enterprise," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 1–12, 2025, doi: 10.1109/TASE.2024.3485810.

[29] J. Li, Q. Liu, C. Wang, and X. Li, "A disjunctive graph-based Metaheuristic for flexible job-shop scheduling problems considering fixture shortages in customized manufacturing systems," *Robot. Comput.-Integr. Manuf.*, vol. 95, Oct. 2025, Art. no. 102981, doi: 10.1016/j.rcim.2025.102981.

[30] M. Liu, J. Lv, S. Du, Y. Deng, X. Shen, and Y. Zhou, "Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimisation," *Comput. Ind. Eng.*, vol. 188, Feb. 2024, Art. no. 109903, doi: 10.1016/j.cie.2024.109903.

[31] D.-K. Lee, J.-H. Shin, and D.-H. Lee, "Operations scheduling for an advanced flexible manufacturing system with multi-fixturing pallets," *Comput. Ind. Eng.*, vol. 144, Jun. 2020, Art. no. 106496, doi: 10.1016/j.cie.2020.106496.

[32] J.-H. Shin, J.-M. Yu, H.-H. Doh, H.-W. Kim, and D.-H. Lee, "Batching and scheduling for a single-machine flexible machining cell with multi-fixturing pallets and controllable processing times," *Int. J. Prod. Res.*, vol. 58, no. 3, pp. 863–877, Feb. 2020, doi: 10.1080/00207543.2019.1602742.

[33] H.-H. Doh, J.-M. Yu, Y.-J. Kwon, D.-H. Lee, and M.-S. Suh, "Priority scheduling for a flexible job shop with a reconfigurable manufacturing cell," *Ind. Eng. Manage. Syst.*, vol. 15, no. 1, pp. 11–18, Mar. 2016, doi: 10.7232/iems.2016.15.1.011.

[34] D.-G. Sim and D.-H. Lee, "A two-stage solution approach for scheduling flexible manufacturing systems with multi-fixturing pallets," *Flexible Services Manuf. J.*, vol. 37, no. 4, pp. 1251–1275, Dec. 2025, doi: 10.1007/s10696-024-09567-5.

[35] Y. Zhou, S. Du, M. Liu, and X. Shen, "Machine-fixture-pallet resources constrained flexible job shop scheduling considering loading and unloading times under pallet automation system," *J. Manuf. Syst.*, vol. 73, pp. 143–158, Apr. 2024, doi: 10.1016/j.jmsy.2024.01.010.

[36] H. Yu, K. Gao, N. Wu, M. Zhou, P. N. Suganthan, and S. Wang, "Scheduling multiobjective dynamic surgery problems via Q-learning-based meta-heuristics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 6, pp. 3321–3333, Jun. 2024, doi: 10.1109/TSMC.2024.3352522.

[37] S. Ye and T.-M. Bu, "A self-learning Harris hawks optimization algorithm for flexible job shop scheduling with setup times and resource constraints," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 2642–2649, doi: 10.1109/SMC52423.2021.9659113.

[38] W. Bożejko et al., "Optimal solving of a scheduling problem using quantum annealing metaheuristics on the D-wave quantum solver," *IEEE Trans. Syst. Man Cybern., Syst.*, vol. 55, no. 1, pp. 196–208, Jan. 2025, doi: 10.1109/TSMC.2024.3458873.

[39] H. Wang, T. Peng, A. Nassehi, and R. Tang, "A data-driven simulation-optimization framework for generating priority dispatching rules in dynamic job shop scheduling with uncertainties," *J. Manuf. Syst.*, vol. 70, pp. 288–308, Oct. 2023, doi: 10.1016/j.jmsy.2023.08.001.

[40] R. Li, W. Gong, L. Wang, C. Lu, and C. Dong, "Co-evolution with deep reinforcement learning for energy-aware distributed heterogeneous flexible job shop scheduling," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 1, pp. 201–211, Jan. 2024, doi: 10.1109/TSMC.2023.3305541.

[41] Y. Li et al., "Real-time scheduling for flexible job shop with AGVs using multiagent reinforcement learning and efficient action decoding," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 3, pp. 2120–2132, Mar. 2025, doi: 10.1109/TSMC.2024.3520381.

[42] Y. Wang, Y. Han, Y. Wang, X. Wang, Y. Liu, and K. Gao, "Reinforcement learning-assisted memetic algorithm for sustainability-oriented multiobjective distributed flow shop group scheduling," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 4, pp. 2399–2413, Apr. 2025, doi: 10.1109/TSMC.2024.3518625.

[43] Y. Li, Q. Liu, C. Zhang, X. Li, and L. Gao, "Graph-based dual-agent deep reinforcement learning for dynamic human–machine hybrid reconfiguration manufacturing scheduling," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 11, pp. 8729–8741, Nov. 2025, doi: 10.1109/TSMC.2025.3612300.

[44] C. Zhang, P. Li, Z. Guan, and Y. Rao, "A Tabu search algorithm with a new neighborhood structure for the job shop scheduling problem," *Comput. Oper. Res.*, vol. 34, no. 11, pp. 3229–3242, Nov. 2007, doi: 10.1016/j.cor.2005.12.002.

[45] J. Fan, C. Zhang, Q. Liu, W. Shen, and L. Gao, "An improved genetic algorithm for flexible job shop scheduling problem considering reconfigurable machine tools with limited auxiliary modules," *J. Manuf. Syst.*, vol. 62, pp. 650–667, Jan. 2022, doi: 10.1016/j.jmsy.2022.01.014.

[46] J. Xie, X. Li, L. Gao, and L. Gui, "A new neighbourhood structure for job shop scheduling problems," *Int. J. Prod. Res.*, vol. 61, no. 7, pp. 2147–2161, Apr. 2023, doi: 10.1080/00207543.2022.2060772.

[47] Q. Liu, C. Wang, X. Li, and L. Gao, "An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated guided vehicle transportation task," *J. Manuf. Syst.*, vol. 70, pp. 127–136, Oct. 2023, doi: 10.1016/j.jmsy.2023.07.004.

[48] Z. Zhang, X. Li, L. Gao, Q. Liu, and J. Huang, "Tackling dual-resource flexible job shop scheduling problem in the production line reconfiguration scenario: An efficient meta-heuristic with critical path-based neighborhood search," *Adv. Eng. Informat.*, vol. 65, May 2025, Art. no. 103282, doi: 10.1016/j.aei.2025.103282.

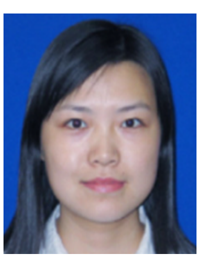
[49] M. Hajibabaei and J. Behnamian, "Flexible job-shop scheduling problem with unrelated parallel machines and resources-dependent processing times: A Tabu search algorithm," *Int. J. Manage. Sci. Eng. Manage.*, vol. 16, no. 4, pp. 242–253, Oct. 2021, doi: 10.1080/17509653.2021.1941368.

[50] S. Tzafestas and A. Triantafyllakis, "Deterministic scheduling in computing and manufacturing systems: A survey of models and algorithms," *Math. Comput. Simul.*, vol. 35, no. 5, pp. 397–434, Nov. 1993, doi: 10.1016/0378-4754(93)90041-r.

[51] G. Zhang, "Improved genetic algorithm for the flexible job-shop scheduling problem," *J. Mech. Eng.*, vol. 45, no. 7, p. 145, 2009, doi: 10.3901/jme.2009.07.145.

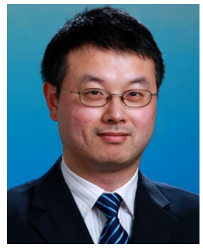
Yulu Zhou was born in Zhejiang, China, in 1999. She received the B.S. degree in industrial engineering from Hefei University of Technology, Hefei, China, in 2022. She is currently pursuing the Ph.D. degree in mechanical engineering with Shanghai Jiao Tong University, Shanghai, China.

Her current research interests are in modeling and solving multiresource-constrained flexible pallet automation system scheduling.



Jun Lv received the Ph.D. degree in business administration from Shanghai University of Finance and Economics, Shanghai, China, in 2008.

She is currently an Associate Professor with the Faculty of Economics and Management, East China Normal University, Shanghai. Her current research interests focus on sustainable operation, production scheduling, and operations research.



Shichang Du (Member, IEEE) received the B.S. and M.S.E. degrees in mechanical engineering from Hefei University of Technology, Hefei, China, in 2000 and 2003, respectively, and the Ph.D. degree in industrial engineering and management from Shanghai Jiao Tong University, Shanghai, China, in 2008.

From 2006 to 2007, he was with the University of Michigan, Ann Arbor, MI, USA, as a Visiting Scholar. He is currently a Professor with the Department of Industrial Engineering and Management, Shanghai Jiao Tong University. He has authored/co-authored 80 articles in journals, including IEEE Transactions on Automation Science and Engineering and *IISE Transactions*. He serves for several journals as an Area Editor, including Computers and Industrial Engineering, Flexible Services and Manufacturing Journal, Precision Engineering, Measurement, Journal of Intelligent Manufacturing, and Computers in Industry. His research has been supported by the National Natural Science Foundation of China, the Ministry of Science and Technology of China, and Shanghai Association for Science and Technology. His current research interests focus on quality control and smart manufacturing methods.

Xiaoxiao Shen (Member, IEEE) received the B.S. degree from the College of Information Management, Nanjing Agricultural University, Nanjing, China, in 2019. She is currently pursuing the Ph.D. degree in mechanical engineering with Shanghai Jiao Tong University, Shanghai, China.

Her current research interests include analysis, modeling, and decision-making optimization for operations management problems in manufacturing systems and service systems. Her technical specialties include mathematical modeling, stochastic optimization, exact optimization algorithms, intelligent optimization algorithms, statistical modeling analysis, and simulation optimization.

Dr. Shen is a member of the Operations Research Society of China and a reviewer of *Computers and Industrial Engineering Journal*.

Molin Liu received the B.E. degree in industrial engineering from Shanghai Jiao Tong University, Shanghai, China, in 2021, and the M.E. degree in industrial engineering and management from Shanghai Jiao Tong University and Centrale Supélec, Gif-sur-Yvette, France, in 2024. He is currently pursuing the Ph.D. degree in technology and operations management with INSEAD, Fontainebleau, France.

His research interests include data-driven optimization, revenue management and pricing, sustainable operations management, and decision-making under uncertainty.