

Article

Transportation-Integrated Flexible Job Shop Scheduling with a Shared Buffer

Xin Liu, Yuangang Wang * , Hongli Liu, Haocheng Zhao and Lin Zhang 

School of Mechanical Engineering, Dalian University, Dalian 116622, China; liuxin@s.dlu.edu.cn (X.L.); 18040140380@163.com (H.L.); zhc1310149774@163.com (H.Z.); lin_zhang@nuaa.edu.cn (L.Z.)

* Correspondence: wygang2000@163.com

Abstract

In flexible job shop scheduling, industrial robots undertake both workpiece transportation and loading/unloading operations. Equipping each machine with dedicated buffers tends to increase transportation workload and further intensify transport bottlenecks. Shared buffers are therefore introduced to temporarily store workpieces and relieve congestion in the production process. This paper establishes a transport-integrated flexible job shop scheduling model with shared buffer constraints, which minimizes makespan, total energy consumption, and machine load range simultaneously. Correspondingly, an enhanced non-dominated sorting genetic algorithm II (ENSGA-II) is developed to achieve better solution performance. A time-window-based path-planning decoding scheme is constructed to address buffer constraints and transportation conflicts in the coordinated production and transportation process. In parallel, four initialization rules are designed to improve the quality and diversity of the initial population, and a variable neighborhood search algorithm (VNS) is embedded to enhance the local exploitation ability of the proposed algorithm. The performance of the presented method is evaluated through two groups of numerical experiments. The first group is carried out on extended benchmark instances. Comparisons with the conventional Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Multi-Objective Particle Swarm Optimization algorithms (MOPSO) validate the efficacy of the proposed strategies and demonstrate the superiority of ENSGA-II in both solution quality and computational efficiency. Experimental results on real-world cases further illustrate that the proposed method can effectively solve the integrated scheduling problem in flexible manufacturing systems where industrial robots are employed as the main transport resources.

Keywords: flexible job shop; transport robot scheduling; shared buffer; multi-objective optimization



Academic Editor: Vasilis K. Oikonomou

Received: 14 April 2026

Revised: 2 June 2026

Accepted: 10 June 2026

Published: 16 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The manufacturing sector is the cornerstone of the national economy, serving as both the foundation for national development and the engine of economic prosperity [1]. Strengthening the manufacturing industry is widely recognized as a strategic pathway for enhancing China's comprehensive national strength and establishing a modern socialist industrial powerhouse. With the progressive integration of emerging technologies—including the Internet of Things (IoT), artificial intelligence (AI), big data analytics, machine vision, virtual simulation, and 5G communication—with traditional manufacturing processes, the sector is undergoing substantial transformation [2]. This integration has introduced

novel concepts, production models, and technological applications, thereby promoting the digitalization, intelligentization, and diversification of manufacturing systems [3]. In this context, the convergence of smart manufacturing and Industry 4.0 has positioned unmanned production, flexible manufacturing, and green, low-carbon operations as key priorities for the transformation and upgrading of the discrete manufacturing sector [4]. Within this framework, flexible job shop scheduling (FJSP) has emerged as a critical methodology to support the modernization of production systems and facilitate the transition toward smart manufacturing in China.

Efficient production management and scientifically guided production scheduling have become critical determinants of competitiveness within the manufacturing industry [5]. Serving as the central mechanism for resource allocation and task sequencing, production scheduling directly influences manufacturing efficiency, utilization of production resources, and overall operational costs [6,7]. This influence is particularly pronounced in high-mix, low-volume production environments, where the degree of scheduling optimization substantially affects both system-level operational performance and comprehensive economic outcomes [6]. Despite its significance, conventional research on FJSP has predominantly concentrated on machine selection and operation sequencing, primarily addressing scheduling challenges within the machining stage. Consequently, these studies often overlook the coordinated decision-making required across machining, transportation, and storage operations, limiting the practical applicability of traditional FJSP approaches in complex manufacturing systems [8].

In practical manufacturing environments, industrial robots play a vital role in workpiece transportation and loading/unloading, and their efficiency directly affects machine continuity, resource utilization, and overall production performance. Accordingly, incorporating transportation resources into flexible job shop scheduling has become an important research direction in recent years. Berterotti et al. proposed the flexible job shop scheduling problem with transportation resources (FJSPT) model, which integrates machine assignment, operation sequencing, vehicle assignment, and routing decisions within a unified framework, thereby demonstrating the necessity of explicitly modeling transportation resources [9]. Luan et al. developed a multi-objective energy-efficient scheduling model and proposed ENSGA-II to improve population diversity and obtain desirable trade-off solutions [10]. Sun et al. incorporated transportation and setup times into a multi-objective model and proposed a hybrid many-objective evolutionary algorithm (HMEA) to optimize makespan, machine load, and earliness/tardiness penalties [11]. Meng et al. studied FJSP with limited automated guided vehicles (AGV) and proposed an improved genetic algorithm to minimize makespan [12]. Zhang et al. further took transportation time, setup time and energy consumption into consideration, and proposed an evolutionary algorithm integrated with reinforcement learning to solve the model [13]. In addition, Xu et al. and Zhang Guohui et al. investigated multi-objective green scheduling problems involving AGVs or material-handling robots and improved solution quality and convergence efficiency through heuristic initialization, local search, and reinforcement learning [14,15]. Overall, existing studies indicate that explicit transportation-resource modeling, multi-objective optimization, and intelligent algorithm design have become key trends in transportation-integrated flexible job shop scheduling.

Nevertheless, most existing studies still assume unlimited buffer capacity and pay limited attention to the storage restrictions faced by workpieces after they leave machining equipment. In practice, limited buffer capacity may lead to workpiece accumulation, machine blocking, and logistics congestion, thereby increasing handling frequency, aggravating path conflicts, reducing equipment utilization, and destabilizing workshop takt time. Therefore, incorporating buffer-capacity constraints into flexible job shop schedul-

ing has become an important direction for coordinating machining, transportation, and storage. Kasapidis et al. developed a unified multi-resource framework integrating finite-capacity machine buffers, Work-in-Progress (WIP) buffers, tools, and shared resources, and proposed a Constraint Programming (CP) model and a CP-based Adaptive Large Neighborhood Search (ALNS-CP) algorithm [16]. Zhang et al. addressed a multi-resource job shop scheduling problem with buffer-capacity constraints and proposed an improved salp swarm algorithm (ISSA) to minimize makespan [17]. Kosuke et al. studied scheduling with AGVs, shared buffers, and collaborative assembly using a genetic algorithm [18]. Aschauer et al. considered bidirectional material flow, blocking, and no-wait constraints and developed a recursive scheduling algorithm combined with tabu search [19]. Sun et al. investigated blocking-constrained distributed hybrid flow shop scheduling with the objectives of minimizing total tardiness and energy consumption [20]. Wen and Guan studied a finite-buffer hybrid flow shop with transportation and energy-consumption considerations and solved it using an energy-saving strategy and lion swarm optimization [21]. Overall, these studies show that buffer constraints not only reshape production rhythm and blocking propagation but are also closely coupled with transportation scheduling, energy consumption, and multi-resource coordination. Table 1 shows a comparison of recent related research.

Table 1. Comparison of related studies on transportation-integrated and buffer-constrained flexible job shop scheduling.

Study	Assumptions	Objectives	Main Constraints Considered	Solution Approach	Difference from This Study
Berterottière et al. [9]	FJSP with transportation resources	Makespan	Machine assignment, operation sequencing, vehicle assignment, routing	Mathematical modeling	Transportation modeled explicitly, no shared buffer or industrial robots
Luan et al. [10]	Energy-efficient FJSP	Makespan, energy	Machine assignment, operation sequencing	ENSGA-II	Energy-aware, no shared-buffer, no robot conflict avoidance
Sun et al. [11]	FJSP with transportation & setup times	Makespan, machine load, earliness/tardiness	Transportation time, setup time, machine assignment, operation sequencing	Hybrid Many-Objective Evolutionary Algorithm	Considers setup & transport, no shared-buffer or robot loading/unloading
Meng et al. [12]	FJSP with limited AGVs	Makespan	Machine assignment, operation sequencing, AGV constraints	Improved Genetic Algorithm	Focused on AGV transport, no shared-buffer, no integrated robot coordination
Zhang et al. [13]	Energy-conscious FJSP with transportation & setup	Makespan, energy	Machine assignment, transportation time, setup, energy	EA + Reinforcement Learning	Considers energy & transport, lacks shared-buffer coordination & industrial robot dual role
Xu et al. [14]	Multi-objective job shop & material handling robots	Makespan, energy	Machine assignment, robot transport, operation sequencing	Reinforcement Learning-guided metaheuristics	Uses AGV or robot transport, lacks integrated decoding & VNS local search
Zhang G. et al. [15]	Multi-objective green scheduling with robots	Makespan, energy	Machine assignment, robot assignment	Heuristic initialization, local search, RL	Focused on energy, lacks shared-buffer & integrated FJSP-TRB modeling
Kasapidis et al. [16]	Multi-resource FJSP with finite buffers	Makespan	Finite machine buffers, WIP buffers, tools, shared resources	CP-based ALNS	Multi-resource, finite buffers, but no industrial robot transportation & decoding framework
Zhang B. et al. [17]	Multi-resource job shop with buffer constraints	Makespan	Buffer capacity, machine resources	Improved Salp Swarm Algorithm	Considers buffers, no shared-buffer & industrial robot transportation integration
Inoue & Aoyama [18]	Assembly line scheduling with human–robot collaboration	Makespan	AGV transport, shared buffers, assembly constraints	Genetic Algorithm	Shared buffers considered, but not energy-aware FJSP with industrial robots

Table 1. *Cont.*

Study	Assumptions	Objectives	Main Constraints Considered	Solution Approach	Difference from This Study
Aschauer et al. [19]	FJSP with blocking and no-wait	Makespan	Blocking, no-wait constraints, bidirectional material flow	Recursive + Tabu Search	Captures blocking, lacks shared-buffer temporary storage & robot transport coordination
Sun et al. [20]	Distributed hybrid flow shop with blocking	Total tardiness, energy	Blocking, machine assignment, transport	Deep RL multi-objective scheduling	Considers distributed flow shop & energy, no integrated FJSP-TRB shared-buffer model
Wen & Guan [21]	Hybrid flow shop with limited buffers & transport	Makespan, energy	Limited buffers, transport	Lion Swarm Optimization	Hybrid flow shop, buffers considered, differs from flexible job shop w/shared buffer & industrial robots
This study	Transportation-integrated flexible job shop scheduling with a shared buffer; industrial robots perform both workpiece transportation and loading/unloading operations	Makespan, total energy consumption, and machine load range.	Machine assignment, operation sequencing, transport robot assignment, shared-buffer transfer, time-window constraints, and robot conflict avoidance	ENSGA-II with three-layer encoding, hybrid initialization, time-window-based decoding, chromosome repair, external archive, and VNS.	Jointly models machining, robot transportation, and shared-buffer rescheduling; explicitly considers industrial robot loading/unloading and conflict avoidance; provides energy-aware Pareto scheduling solutions balancing production efficiency, energy consumption, and machine load.

As shown in Table 1, existing studies have made important progress in transportation-integrated flexible job shop scheduling, energy-aware scheduling, and buffer-constrained scheduling. However, several limitations remain. First, most existing studies focus on dedicated transport resources such as AGVs, while limited attention has been paid to industrial robots performing both transportation and loading/unloading tasks. Second, buffer-related studies mainly consider local or general finite-capacity buffers, whereas shared-buffer-based storage and reschedule mechanisms remain underexplored. Third, transport conflict detection, path coordination, and robot task decoding under time-window constraints in shared-buffer environments have not been sufficiently investigated.

To address these issues, this paper studies a multi-objective energy-efficient flexible job shop scheduling problem with shared buffers and industrial robots for inter-operation transportation. Compared with dedicated buffers assigned to individual machines, a shared buffer is more suitable for the manufacturing system considered in this study, because workpiece transportation and loading/unloading operations are jointly performed by industrial robots. With a dedicated buffer layout, workpieces are temporarily stored beside individual machines, forcing robots to repeatedly perform picking, placing, and transport operations between buffers. This frequent handling increases robot workload and utilization, while also exacerbating transportation conflicts and logistics congestion [22]. In contrast, a shared buffer adopts a centralized temporary storage mode, which can reduce unnecessary transportation among different buffer areas and improve the coordination among machining, transportation, and storage resources [23,24]. Nevertheless, the use of a shared buffer also increases scheduling complexity. Since multiple workpieces may compete for the same buffer space and robot services, buffer utilization must be jointly coordinated with machine assignment, operation sequencing, robot scheduling, and path-conflict resolution. Therefore, it is necessary to develop an integrated scheduling method that simultaneously considers shared-buffer constraints and transportation conflict issues.

Therefore, this study investigates a multi-objective, energy-efficient flexible job shop scheduling problem with shared buffers, in which inter-operation transportation is performed by industrial robots. In the considered system, if a workpiece cannot directly enter its next machine after completing the current operation, it is transported to a shared buffer

for temporary storage and then delivered to the target machine at an appropriate time. Meanwhile, path conflicts during robot handling operations are detected and resolved under time-window constraints. Accordingly, a scheduling model with transportation and shared-buffer constraints is established, and ENSGA-II is proposed to optimize makespan, total energy consumption, and machine load balancing.

2. Problem Description and Mathematical Modeling

2.1. Problem Description

The flexible job shop scheduling problem with transportation and resource buffers (FJSP-TRB) can be described as follows. Consider a manufacturing system consisting of n workpieces, denoted as $J = \{J_1, J_2, \dots, J_n\}$, each requiring s operations $Q = \{Q_1, Q_2, \dots, Q_s\}$, with the number of operations varying across workpieces. The system contains m processing machines, $M = \{M_1, M_2, \dots, M_m\}$, where each operation may be executed on multiple alternative machines. Both processing time and unit energy consumption vary depending on the chosen machine. To support inter-process workpiece transfer, the system is equipped with r transport robots, each capable of performing only one transportation task at a time. A shared vertical silo functions as a centralized intermediate storage node within the workshop, accommodating temporary storage of workpieces during process transitions and thereby enabling coordinated execution of machining, transportation, and buffering operations. In this study, the shared vertical silo is adopted instead of independent machine-side buffers. The reason is that the transport resources are industrial robots rather than simple conveyor-based transport devices. These robots are required to perform both transportation and loading/unloading operations. If each machine is equipped with an independent buffer, a workpiece may need to be repeatedly transferred between different local buffers before entering the next operation. This would increase the number of robot clamping, placement, and loading/unloading operations, thereby increasing robot utilization pressure and the probability of transport conflicts. Therefore, the shared buffer is introduced as a centralized intermediate storage node to reduce redundant handling operations and to improve the overall coordination of the production and transportation processes. Figure 1 illustrates a representative scheduling layout for a workshop equipped with two transport robots, highlighting the interaction between processing machines, transport robots, and the shared buffer.

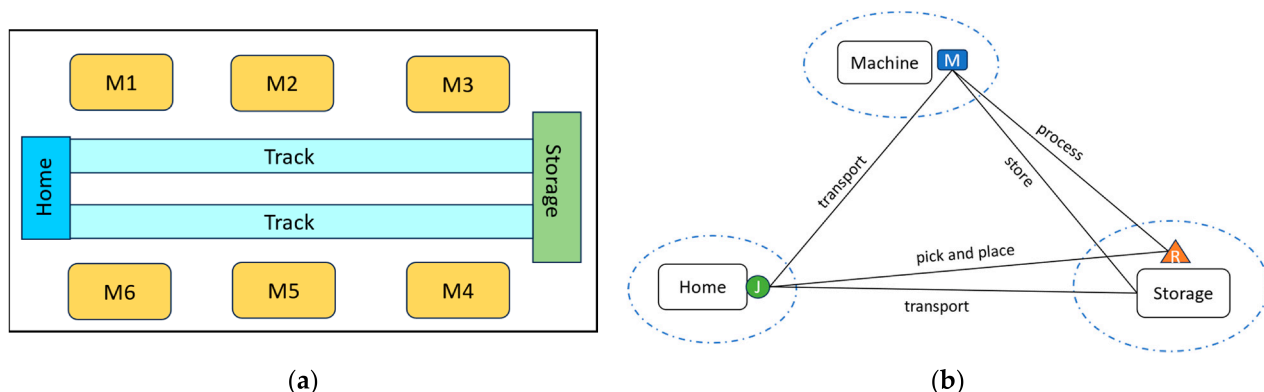


Figure 1. Overall Layout of the Workshop Scheduling System. (a) Flexible Workshop Equipment Layout; (b) Flowchart of Workshop Scheduling.

In this study, an integrated scheduling model is developed for the transport robots, explicitly incorporating the shared buffer zone. The proposed model seeks to optimize multiple objectives concurrently, such as minimizing the maximum completion time (makespan) and the total energy consumption of both processing and transport operations, and the

variation in machine load distribution, thereby achieving balanced and energy-efficient scheduling within the workshop.

2.2. Model Assumptions

To facilitate the modeling of the FJSP-TRB problem and to emphasize the primary coupling relationships among machining resources, transport resources, and the shared vertical storage mechanism, the following assumptions are made:

- (1). At time zero, all machining machines and transport robots are available for operation.
- (2). All workpieces to be processed are initially stored in the vertical storage rack, which serves as the common starting point for all workpieces.
- (3). All transport robots are initially positioned at their predefined starting locations along the transport tracks.
- (4). Sequential dependencies exist between the operations of a single workpiece, whereas different workpieces are mutually independent.
- (5). Each machining machine is capable of processing only one workpiece at a time.
- (6). Each transport robot can carry only one workpiece at any given time.
- (7). The first operation of each workpiece must be transported by a transport robot.
- (8). No transportation is needed when two consecutive operations of the same workpiece are assigned to the same machine.
- (9). The vertical storage rack functions as the system's unified buffer area.
- (10). Workpiece processing cannot be interrupted once initiated.
- (11). The operation of a transport robot cannot be interrupted during material handling.
- (12). A workpiece remains on its current processing machine until it is picked up by a transport robot upon completion of the operation.
- (13). Workpieces stored in the buffer zone follow a first-in, first-out (FIFO) principle.
- (14). Equipment failures, workpiece deterioration, and urgent job insertions are not considered in the model. The production system is assumed to be deterministic, thereby simplifying the initial modeling work and algorithm development process.
- (15). The system's energy consumption consists of two components: the energy used by the processing machines and the energy consumed by the transport robots.
- (16). This model assumes that the transport robots operate at a constant speed and does not consider battery charging cycles.

2.3. Notation of Parameters

The symbols and their definitions used in this paper are provided in Table 2.

Table 2. Notation and definitions.

Symbol	Definition
J	Set of workpieces, $J = \{J1, J2, \dots, Jn\}$
M	Set of machining centers, $M = \{M1, M2, \dots, Mz\}$
R	Set of transport robots, $R = \{R1, R2, \dots, Rh\}$
i, j	Workpiece and operation indices
z	Machining center indices
h	Transport robot indices
O_{ij}	Operation j of workpiece i
M_{ij}	Set of available machines for operation O_{ij}
t_i^s	Start time of workpiece J_i
t_i^e	Finish time of workpiece J_i
T_{ijz}^s	Start time of operation O_{ij} on machine M_z
T_{ijz}^e	End time of operation O_{ij} on machine M_z

Table 2. Cont.

Symbol	Definition
T_{ijz}	Processing time of operation O_{ij} on machine M_z
T_{ijz}^w	Idle time of machine M_z
T_{ijh}	Task for the robot to transport operation O_{ij} to the corresponding machine
TT_{ab}	Transportation time between machines M_a and M_b
T_{ij}^{trans}	Transfer time from operation O_{ij} to the next operation
$LT_{h,ij}^s$	Start time of robot empty run
$LT_{h,ij}^e$	End time of robot empty run
$LT_{h,ij}$	Duration of robot empty run
$WT_{h,ij}^s$	Start time of robot loaded run
$WT_{h,ij}^e$	End time of robot loaded run
$WT_{h,ij}$	Duration of robot loaded run
$WT_{h,ij}^w$	Idle time of robot R_h
MP_z^w	Idle energy consumption per unit time for machine M_z
MP_z^p	Processing energy consumption per unit time for machine M_z
EP_h^s	Unit energy consumption of robot R_h in a stationary state
EP_h^e	Unit energy consumption of robot R_h in an empty state
EP_h^l	Unit energy consumption of robot R_h in a loaded state
EP_h^a	Energy consumption of robot R_h in an active state
V_z	Location of the transport robot
V_{ij}	Location of process O_{ij}
$V_{i'j'}$	Location of the preceding process $O_{i'j'}$
T^p	Time for robot to arrive at node p
T_{zu}^r	Transportation time from machine M_z to the hopper
T_{ijz}^w	Waiting time for process O_{ij} at machine M_z
T_{iju}^w	Waiting time for process O_{ij} at the hopper
(x_p, y_p)	Coordinates of node p
X_{ijz}	1 if process O_{ij} is being processed by machine M_z ; 0 otherwise
$K_{i'j'ijz}$	1 if process $O_{i'j'}$ is processed by machine M_z before process O_{ij} ; 0 otherwise
Y_{ijh}	1 if process O_{ij} is transported by robot R_h ; 0 otherwise
$G_{i'j'ijh}$	1 if process $O_{i'j'}$ is transported by robot R_h before process O_{ij} ; 0 otherwise
Z_{ijz}	1 if process O_{ij} is placed in the buffer after processing on machine M_z ; 0 otherwise
M	A sufficiently large number

2.4. Robotic Transport Model in a Shared Buffer Area

The shared-buffer strategy has both advantages and limitations compared with independent buffers dedicated to individual machines. For independent buffers, each machine has its own local storage space, which can simplify local machine-level storage management. However, this configuration may lead to uneven buffer utilization. Some machine-side buffers may become congested while others remain idle. More importantly, in the considered workshop, industrial robots undertake both transportation and loading/unloading operations. If independent buffers are used, robots may need to perform additional clamping, unloading, and reloading operations when workpieces are transferred among different machine-side buffers. This increases the handling frequency of the robots and may aggravate transportation bottlenecks, especially when several machines simultaneously request robot service. In contrast, the shared buffer provides a centralized temporary storage area for multiple machines. It can improve buffer-space utilization, reduce unnecessary repeated transfers among local buffers, and support more flexible robot dispatching. When the next machine of a workpiece is not immediately available, the workpiece can be temporarily stored in the shared buffer and then transported to the target machine at a suitable time. This helps release the current machine earlier and improves the coordination among processing, transportation, and storage operations. Nevertheless, the shared-buffer strategy also has potential disadvantages. Since multiple workpieces may compete for the same

shared buffer and the same transport robots, buffer access conflicts and additional waiting time may occur. Therefore, the shared buffer cannot be treated as a simple storage area; it must be integrated into the scheduling model together with robot assignment, path planning, time-window constraints, and conflict-avoidance rules. This is why the proposed decoding strategy explicitly considers shared-buffer transfer decisions and robot transportation conflicts.

2.4.1. Workpiece Buffer Path Planning

In transportation-integrated flexible job shop scheduling under shared buffer constraints, this paper considers the following three transportation schemes. The three transportation schemes are illustrated in Figure 2.

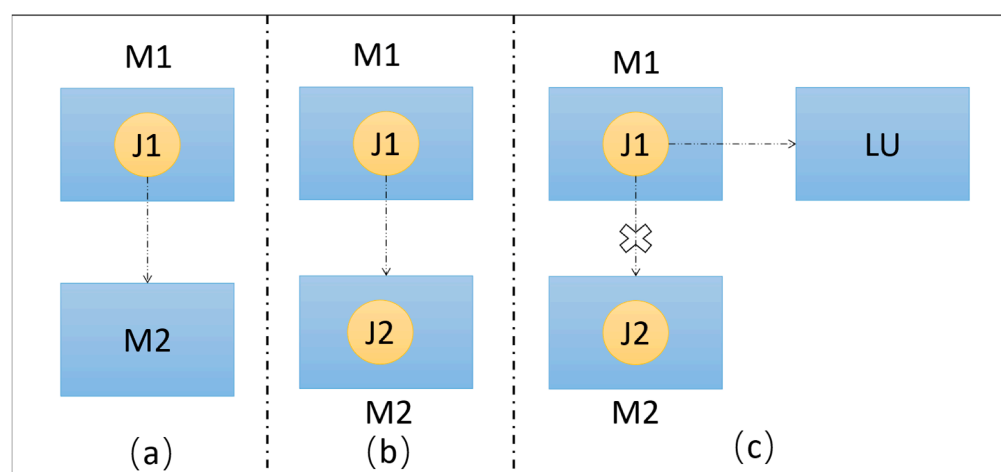


Figure 2. Buffer Path Planning Layout. (a) Direct Transfer; (b) Delayed Transfer at Current Machine; (c) Cannot be processed directly after being stored in the buffer.

1. Direct Transfer: If the target machine is readily available and a transport robot is present, the workpiece is directly transferred from the current machine to the machine designated for the next operation.
2. Delayed Transfer at Current Machine: If the target machine is available but no transport robot is currently free, the workpiece remains at the current machine until a transport robot becomes available. Alternatively, if the target machine is unavailable but the earliest feasible processing time at the current location occurs before the earliest time required for temporary storage in the buffer, the workpiece continues waiting at the current machine.
3. Buffer Storage Transfer: If the target machine is temporarily unavailable and the earliest available processing time of the workpiece at its current location is later than the earliest available time after transporting it to the buffer, the workpiece will be transferred to the shared vertical storage for temporary storage, awaiting the next picking time.

2.4.2. Robot Collision-Avoiding Path Planning

In scheduling multiple transport robots within a workshop, potential collisions must be addressed, especially those arising from interference between robotic arms during workpiece pick-up and placement along the seventh axis. The typical conflict conditions considered in the robot collision-avoidance mechanism are shown in Figure 3. These conflicts fall into two scenarios:

1. **Static Avoidance Conflicts:** If a transport robot R_1 needs to pick up a workpiece from a machine located on the opposite side, and another robot R_2 is positioned such that it could obstruct R_1 while remaining stationary, R_2 must first move to a predefined safe distance before R_1 initiates the pick-up operation. This maneuver prevents collision between the robotic arms.
2. **Dynamic Priority Conflicts:** If two transport robots, R_1 and R_2 , simultaneously require access to opposite-side machines at the same location, priority is assigned to the robot whose workpiece has the longer subsequent processing time. The other robot, with the shorter subsequent processing time, must yield by moving to a safe distance and waiting until the operation is completed by the prioritized robot.

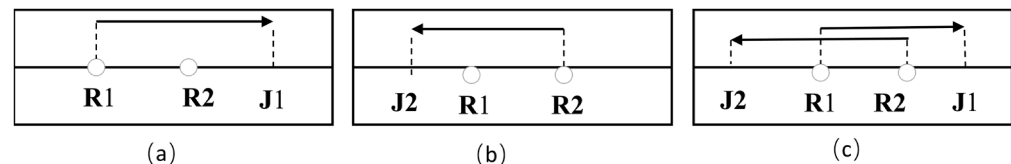


Figure 3. Conflict-Aware Path Planning Layout. (a) Conflict condition: the traveling route of R_1 overlaps and crosses R_2 's path; (b) conflict condition: the traveling route of R_2 overlaps and crosses R_1 's path; (c) conflict condition: paths of two vehicles cross each other.

By implementing this conflict resolution mechanism, robotic arm interference in multi-robot collaborative operations is effectively reduced, enhancing both safety and efficiency in workshop scheduling. This approach ensures stability in the workpiece processing sequence and maintains the overall production rhythm. Each workpiece sequentially undergoes transportation from the vertical silo (or previous machine) to the designated machine, processing, inter-machine transfer, waiting due to conflicts, and transportation back to the silo after machining. The shared vertical storage serves as a staging area when the target machine is unavailable, and transport robots manage the timely movement of workpieces to the next operation. This makes transport a critical element of the production workflow, significantly impacting system performance alongside machine allocation and sequencing.

2.5. Energy Consumption Model

1. Total Energy Consumption

The total energy consumption of the system (SE) primarily originates from two components: the machining stage and the transport stage. To achieve energy-efficient scheduling, both components must be considered comprehensively, and overall energy consumption should be minimized through optimized scheduling.

$$SE = ME + TE \quad (1)$$

where ME represents the total energy consumed by the machining machines, and TE denotes the total energy consumed by the transport robots.

2. Machining Machine Energy Consumption

The machining machine energy consumption (ME) includes both the energy used during active processing and the energy consumed during idle (standby) states throughout the production cycle:

$$ME = \sum_{i=1}^n \sum_{j=1}^s \sum_{z=1}^m \left(ME_{ijz}^p + ME_{ijz}^w \right) \quad (2)$$

(1) Processing Energy Consumption (ME_{ijz}^p)

This component reflects the energy consumed by the machine M_z while machining a workpiece during the operation Q_j . It depends on the operation's machining time and the machine's rated processing power:

$$ME_{ijz}^p = \sum_{z=1}^m \sum_{i=1}^n \sum_{j=1}^s T_{ijz} \times MP_z^p \quad (3)$$

$$T_{ijz} = T_{ijz}^e - T_{ijz}^s \quad (4)$$

where T_{ijz}^s and T_{ijz}^e denote the start and end times of the operation, respectively, and MP_z^p is the machine's power during processing.

(2) Standby Energy Consumption (ME_{ijz}^w)

Standby energy refers to the energy consumed while the machine is idle but remains operational during the scheduling cycle. It depends on the idle duration and the standby power:

$$ME_{ijz}^w = \sum_{z=1}^m \sum_{i=1}^n \sum_{j=1}^s T_{ijz}^w \times MP_z^w \quad (5)$$

$$T_{ijz}^w = \sum_{p=1}^n \sum_{q=1}^s \sum_{z=1}^m \sum_{i=1}^n \sum_{j=1}^s X_{pqz} \times X_{ijz} \times XP_{pqijz} \times (T_{ijz}^s - T_{pqz}^e) \quad (6)$$

where MP_z^w is the standby power of machine M_z , and X_{pqz} , X_{ijz} , XP_{pqijz} are scheduling decision variables and indicators representing machine assignment and operation precedence.

3. Robot Transport Energy Consumption

The transport robot energy consumption (TE) consists of four components: load energy, no-load energy, idle energy, and dynamic energy during manipulation:

$$TE = \sum_{i=1}^n \sum_{j=1}^s \sum_{h=1}^r (TE_{ijh}^e + TE_{ijh}^w + TE_{ijh}^l + TE_h^a) \quad (7)$$

(1) Load Energy (TE_{ijh}^l)

Energy consumed while the robot transports a grasped workpiece between processing stations:

$$TE_{ijh}^l = LT_{ijh} \times EP_{ijh}^l \quad (8)$$

(2) No-Load Energy (TE_{ijh}^e)

Energy consumed when the robot moves without carrying a workpiece, from its initial position, a machine, or the vertical storage rack to the task start location.

$$TE_{ijh}^e = WT_{ijh} \times EP_{ijh}^e \quad (9)$$

(3) Idle Energy (TE_{ijh}^w)

Energy consumed while the robot is waiting and not performing transport tasks

$$TE_{ijh}^w = WT_{ijh}^w \times EP_{ijh}^w \quad (10)$$

(4) Dynamic Energy (TE_h^a)

Energy consumed due to joint rotations and manipulations during workpiece grasping or loading:

$$TE_h^a = N_h \times EP_h^a \quad (11)$$

where LT_{ijh} , WT_{ijh} , WT_{ijh}^w , N_h are the respective operation durations, and EP_{ijh}^l , EP_{ijh}^e , EP_{ijh}^w , EP_h^a are the corresponding energy coefficients for each state.

2.6. Mathematical Model

2.6.1. Objective Function

1. Minimization of Maximum Completion Time

The maximum completion time, or makespan, refers to the latest completion time of all workpieces. A shorter makespan indicates higher production efficiency:

$$f_1 = \min(\max C_i) \quad i = 1, 2, \dots, n \quad (12)$$

where C_j denotes the completion time of workpiece J_j .

2. Minimization of Total Energy Consumption

In addition to minimizing makespan, reducing total energy consumption helps lower operational costs and supports the goal of energy-efficient:

$$f_2 = \min SE \quad (13)$$

3. Minimization of the Range of Machine Loads

Balancing machine loads avoids prolonged idling or localized overloads, thereby improving overall system utilization. The range of machine loads is defined as:

$$f_3 = \max_{z=1..M} L_z - \min_{z=1..M} L_z \quad (14)$$

$$L_k = \sum_{i=1}^n \sum_{j=1}^s \sum_{z=1}^m T_{ij} X_{ijz} \quad (15)$$

where L_z denotes the total workload assigned to machine M_z , T_{ij} is the processing time of operation O_{ij} , and X_{ijz} is a binary variable indicating whether operation O_{ij} is assigned to machine M_z .

2.6.2. Constraints

$$\sum_{z=1}^n X_{ijz} = 1, \quad \forall i, j \quad (16)$$

$$\sum_{z=1}^m y_{ijz} = 1, \quad \forall i, j \quad (17)$$

$$T_{ijz}^s \geq T_{i,j-1,z}^e, \quad \forall i, j, z \quad (18)$$

$$T_{i'j'z}^e + X_{ijz} T_{ijz} \leq T_{ijz}^e + (1 - K_{i'j'ijz}) \cdot M \quad (19)$$

$$T_{ijz}^e = T_{ijz}^s + T_{ijz} \quad (20)$$

$$LT_{ijh}^s + (1 - G_{i'j'ijz}) \cdot M > WT_{i'j'h}^e \quad (21)$$

$$LT_{ijh}^s \geq WT_{i,j-1,h}^e \quad (22)$$

$$LT_{ijh}^e = LT_{ijh}^s + \sum_{i=1}^n \sum_{j=1}^s \sum_{z=1}^m \sum_{z'=1}^m X_{i,j-1,z'} X_{ijz} T_{zz'} \quad (23)$$

$$WT_{ijh}^e = WT_{ijh}^s + \sum_{i=1}^n \sum_{j=1}^s \sum_{z=1}^m \sum_{z'=1}^m X_{i,j-1,z'} X_{ijz} T_{zz'} \quad (24)$$

$$T_{ijz}^e = T_{ijz}^s + T_{ijz} \quad (25)$$

$$T_{ijz}^v = T_{i,j+1,z}^s - T_{ijz}^e - T_{ijtrans} \quad (26)$$

$$T_{i,j+1,z}^s = \begin{cases} T_{ijz}^e + T_{ijz}^v + T_{ijtrans}, & (Z_{ijz} = 0) \\ T_{ijz}^e + T_z^r + T_{z'}^r + T_{iju'}^v, & (Z_{ijz} = 1) \end{cases} \quad (27)$$

Equation (16) states that each process can be executed by only one machine; Equation (17) specifies that each transport task can be carried out by only one transport machine; Equation (18) specifies the processing sequence for two adjacent processes of a job; Equation (19) specifies the priority order of two processes on the same machine; Equation (20) specifies the completion time of a process on a machine; Equation (21) indicates that the current task will only begin transportation after the previous task on the assigned transport machine has been completed; Equation (22) ensures that the start time of an empty run is greater than or equal to the completion time of the previous process; Equation (23) specifies that the completion time of an empty run is the sum of the start time and the transportation time between the two relevant machines; Equation (24) shows that the completion time of a load task is the sum of its start time and the transport time between the two relevant machines; Equation (25) indicates that the processing of the current workpiece can only begin once the previous process is completed and the workpiece has been delivered; Equation (26) indicates that load transport can only begin after the transport machine has arrived and the previous process of the workpiece has been completed; and Equation (27) indicates the start time of the next process.

3. ENSGA-II Algorithm

To address the interdependencies among objectives in multi-objective optimization, this study improves NSGA-II by designing a three-segment encoding scheme tailored to the problem characteristics and combining four initialization methods to enhance population quality. A buffer decoding strategy based on time-window constraints is introduced for transportation planning, and variable-domain search is incorporated to strengthen local search capabilities, thereby improving the overall performance and convergence of the algorithm [25]. The overall workflow of the algorithm is shown in Figure 4.

3.1. Three-Tier Coding Based on Production Processes

Considering the coordinated decision-making requirements for process sequencing, machine allocation, and transportation resource assignment in this study, a three-tiered coding structure has been designed. The first tier corresponds to the process selection layer, the second tier represents the machine allocation layer, and the third tier encodes the transportation resource allocation layer. A schematic illustration of this coding structure is provided in Figure 5.

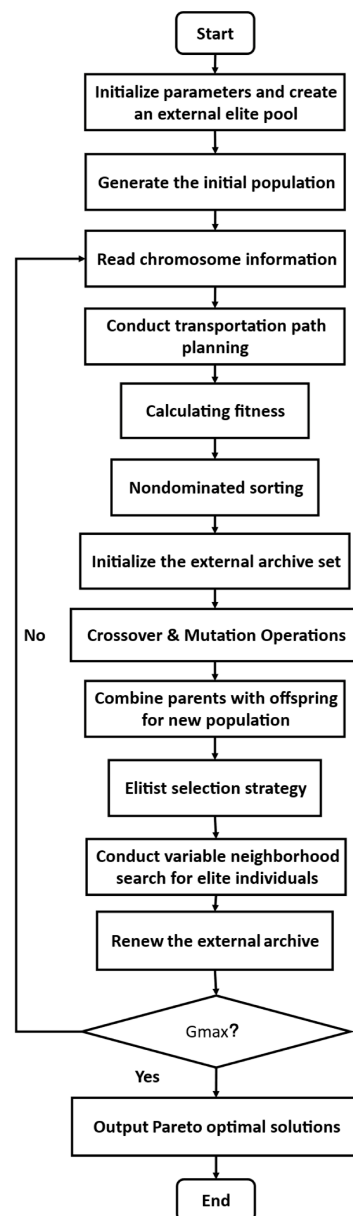


Figure 4. Flowchart of ENSGA-II Algorithm.

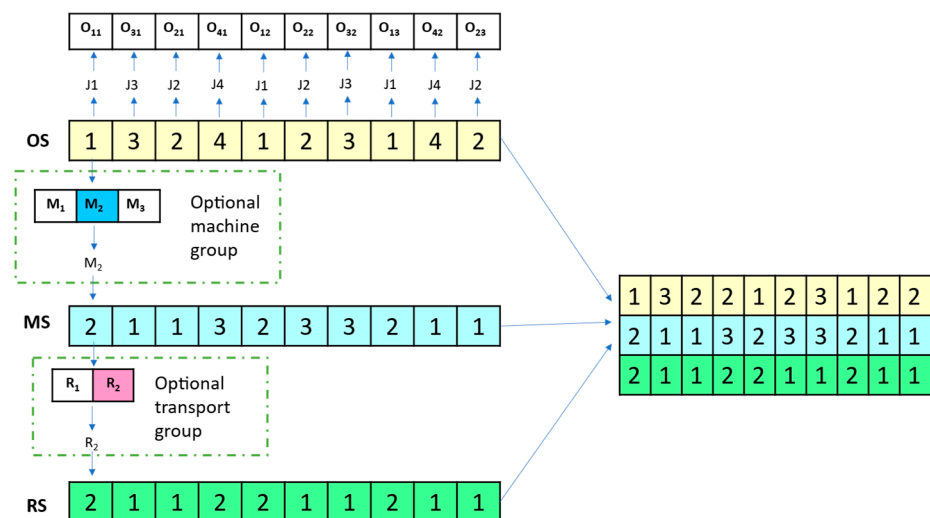


Figure 5. Schematic Example for Three-Hierarchical Encoding Strategy.

3.2. Time-Window Constraint-Based Path Planning Decoding Strategy

Decoding converts the encoded representation into a feasible scheduling plan. To address core scheduling challenges such as transfers via the shared vertical silo, conflicts at robot pick-up points, and the constraint that machines cannot be released until workpieces are retrieved, a time-window constraint-based path decoding strategy is proposed. By incorporating time-window constraints, this approach ensures the feasibility and coordination between machining, robot movements, and buffer utilization, enabling integrated scheduling within the FJSP-TRB framework. The detailed decoding and scheduling process for each workpiece in the FJSP-TRB framework is as follows:

Step 1: Information Retrieval

For the current operation O_{ij} , relevant information is retrieved from the chromosome, including: the assigned processing machine M_a for the current operation, its completion time T_{eijz} , and its location V_{ijk} ; the assigned processing machine M_b for the subsequent operation $O_{i,j+1}$, the completion time T_{emb} of the preceding task on that machine, and its location $V_{i(j+1)k}$; the status and current location of the transport robot V_{ijk} ; and the transportation time T_{ij}^{trans} between the current machine and the next machine.

Step 2: Location Comparison

Determine whether V_{ijk} and $V_{i(j+1)k}$ correspond to the same machine location. If they are identical, proceed directly to Step 5; otherwise, proceed to Step 3.

Step 3: Next-Machine Status and Transfer Time Calculation

Evaluate the status of the next-process machine M_b and compute the workpiece transfer time:

1. Direct Transfer: If $T_{eijz} + T_{ij}^{trans} \geq T_{emb}$, the workpiece arrives at M_b when it is already idle and can be transported directly for processing. Figure 6 illustrates the scenario of direct transfer.

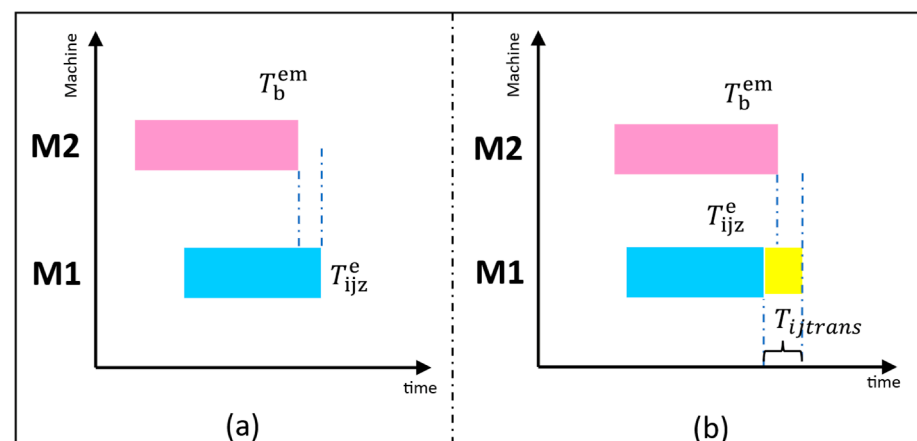


Figure 6. Direct Processing and Loading Layout. (a) $T_{eijz} \geq T_{emb}$; (b) $T_{eijz} + T_{ij}^{trans} \geq T_{emb}$.

2. Delayed Transfer: If $T_{eijz} + T_{ij}^{trans} < T_{emb}$, i.e., M_b is still processing, calculate the earliest possible arrival times for two options. Figure 7 illustrates the two alternative transfer schemes.

Option A: The workpiece remains at the current machine M_a until M_b becomes available and a transport robot is ready; then, it is transported directly to M_b .

Option B: The workpiece is first transported to a shared buffer by a transport robot and subsequently moved to M_b once the machine becomes idle.

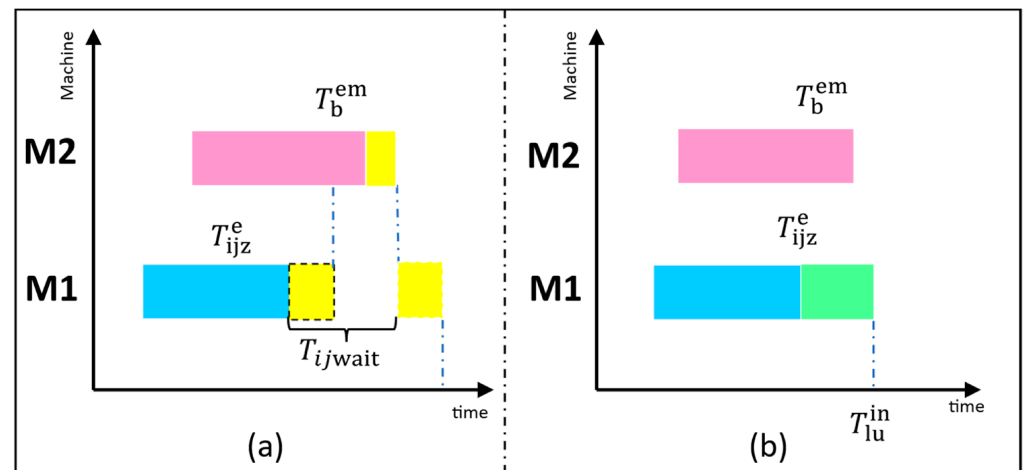


Figure 7. Schematic of Waiting for Processing. (a) Workstation Waiting–Processing Schematic; (b) Buffer Waiting–Processing Schematic.

The workpiece follows the option with the earlier feasible arrival time:

If Option A's arrival time is no later than Option B, the workpiece waits at M_a .

If Option B's arrival time is earlier, the workpiece enters the shared buffer, and the entry time is recorded as $T_{lu}^{in} = T_{ijz}^e + T_{trans(Ma \rightarrow lu)}$.

Step 4: Transport Robot Scheduling

Determine the transport robot's status and compute the dynamic transport time:

If the robot's current location V_{ijkh} matches V_{ijk} , it can immediately begin loading and transporting the workpiece. Alternatively, record the time it takes for the robot to travel from its current location to V_{ijkh} . Upon arrival, the robot loads and transports the workpiece.

Step 5: Machine-Robot Side Determination

Check whether the machining machine and the transport robot are on the same side:

If yes, the transport robot immediately begins loading and transporting the workpiece.

Figure 8 illustrates the schematic of conflict-free direct gripping.

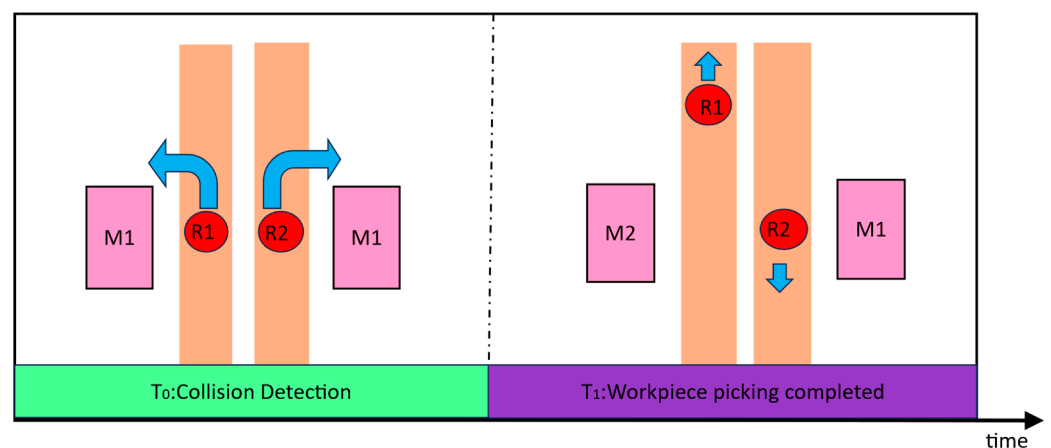


Figure 8. Conflict-free Direct Gripping Schematic.

If no, determine whether conflict resolution is necessary based on the time-window constraints. If a conflict is detected, proceed to Step 6; otherwise, move on to Step 7.

Step 6: Conflict Resolution

Identify the type of conflict and perform the appropriate resolution, updating the transport robot's time window accordingly. Figure 9 illustrates the resolution of a unidirectional conflict. Figure 10 illustrates the resolution of a bidirectional conflict.

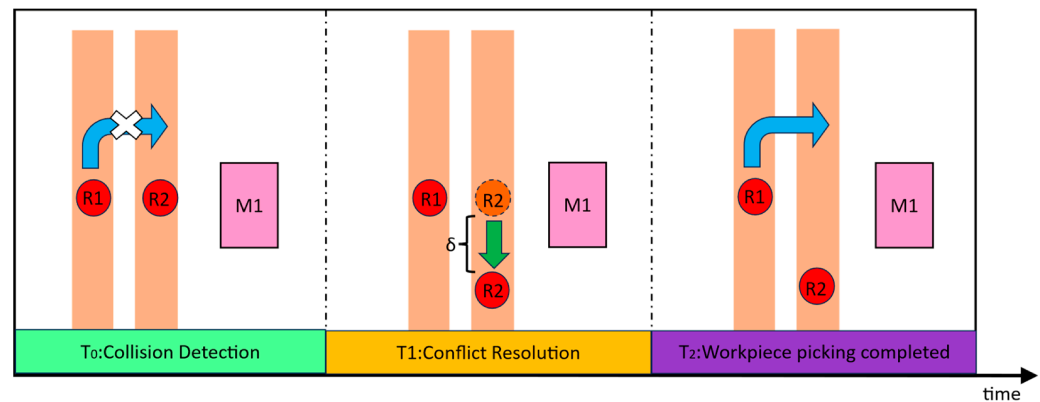


Figure 9. Unidirectional Conflict Resolution.

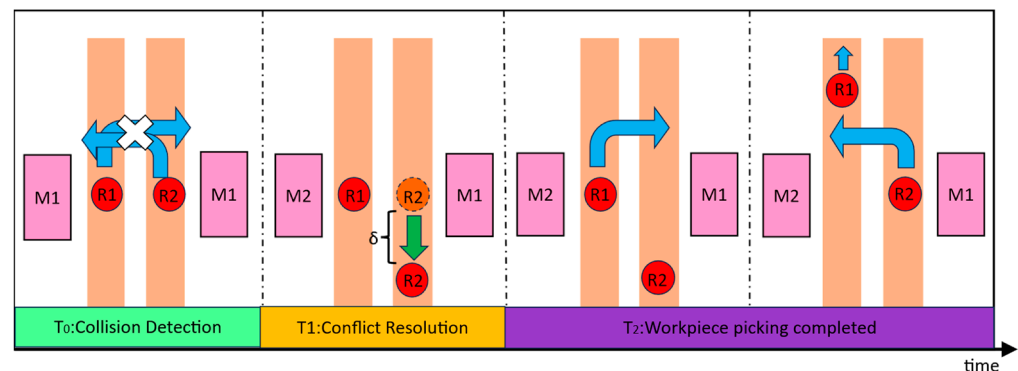


Figure 10. Bidirectional Conflict Resolution.

Step 7: Update Operation Start Time

Calculate the workpiece's arrival time at the next machine and update the actual start time for the operation $O_{i,j+1}$.

Step 8: Iteration and Performance Evaluation

Repeat Steps 2–7 until all operations of all workpieces have been scheduled. Next, the scheduling performance metrics are calculated, including the maximum completion time, total energy consumption, and machine utilization, along with auxiliary metrics such as shared-buffer waiting time.

3.3. Population Initialization

To ensure broad coverage of the solution space and the generation of high-quality feasible solutions in the early stages of the algorithm, a hybrid initialization strategy is employed, combining random generation with heuristic-based methods. The initialization rules are as follows:

1. Fully Random Generation Method: The operation sequence (OS) is generated in a fully random manner. Subsequently, machine selection (MS) and resource selection (RS) are performed according to the OS, with machine and transport robot IDs randomly selected from the set of available resources.
2. Long-Process Priority Generation Method: Processes with longer processing times are more likely to form the critical path and significantly influence the overall makespan. Therefore, in the OS layer, operations with longer processing times are prioritized.

3. **Minimum Processing Time Generation Method:** In the machine assignment layer (MS), the machine with the shortest processing time and the earliest availability is chosen. In the transport resource allocation layer (RS), transport robots with the earliest availability and shortest travel paths are prioritized to minimize waiting times and reduce the total completion cycle.
4. **Transportation Bottleneck Alleviation Priority:** Scheduling is prioritized for workpieces with intensive transportation requirements or longer travel paths. During the RS allocation phase, robots with lighter loads are preferentially assigned to reduce the risk of transport congestion and improve workflow continuity.

3.4. Genetic Operators

The crossover and mutation operators used in the proposed ENSGA-II are standard, well-established operators commonly applied in multi-objective evolutionary algorithms.

3.4.1. Crossover

Crossover operations generate new offspring by combining the genetic information of two or more parent individuals, which enhances population diversity, promotes a thorough search of the solution space, and increases the probability of finding the global optimum. This study adopts a hierarchical crossover strategy: for the operation sequence (OS) layer, a priority operation crossover (POX) is used; for the machine assignment (MS) layer and transport resource assignment (RS) layer, uniform crossover is utilized to further enhance diversity. Figure 11 illustrates the crossover operation.

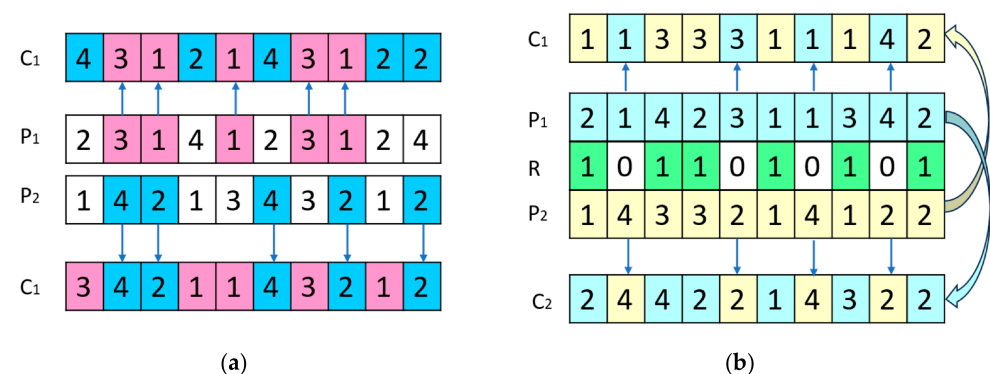


Figure 11. Crossover Operation. (a) The POX crossover operation is performed on the OS layer; (b) the uniform crossover operation is implemented on both the MS layer and RS layer.

3.4.2. Mutation

To further improve population diversity and reduce premature convergence in the evolutionary algorithm, multi-mechanism mutation operations are applied, specifically tailored to the different encoding layers. For the operation sequence layer, exchange mutation is used, whereby the gene sequences at two randomly selected positions are swapped. This strategy allows for local rearrangements in the operation order while preserving the overall feasibility of the sequence. The machine allocation layer and the transport machine layer employ reassignment mutation. A random subset of processes is selected, and their assigned machines or transport robots are replaced with alternative resources from the available pool. This approach diversifies machine and robot assignments, reduces the risk of localized congestion, and facilitates exploration of the solution space. Figure 12 illustrates the mutation operation.

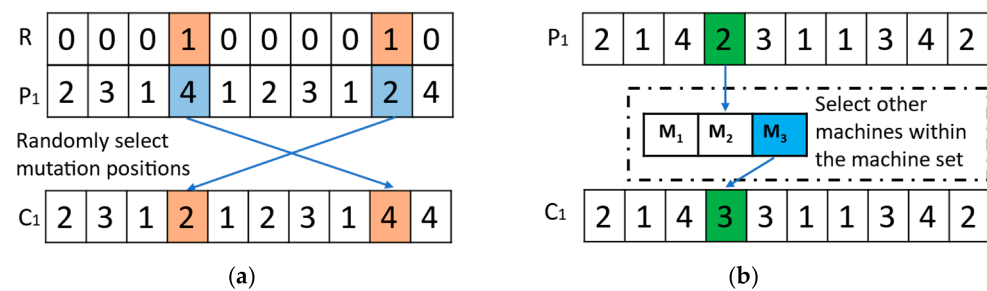


Figure 12. Mutation Operation. (a) Swap Mutation on OS Layer; (b) Reallocation Mutation for MS Layer and RS Layer.

3.4.3. Adaptive Strategy

In genetic algorithms, crossover and mutation operations occur with certain probabilities. The crossover probability P_c and mutation probability P_m determine the balance between convergence and search capability. Early genetic algorithms that adopted fixed probability values often suffered from reduced computational performance and were not suitable for large-scale instances. In this study, the crossover probability P_c and mutation probability P_m are adaptively adjusted during the evolutionary process. When the individuals in the non-dominated solution set tend to be concentrated, the mutation probability is appropriately increased to enhance the exploration capability. Conversely, when the solution set is relatively dispersed, the crossover probability is increased to accelerate convergence.

$$P_c = \begin{cases} P_c^{max} - (P_c^{max} - 0.6) \left(\frac{\bar{f}}{f_{max}} - \frac{f - \bar{f}}{2(f_{max} - \bar{f})} \right), & \bar{f} \geq f \\ P_c^{max}, & \bar{f} < f \end{cases} \quad (28)$$

$$P_c^{max} = \begin{cases} 0.9, & f_{min} \leq \frac{1}{2}f_{max} \\ 0.8, & \frac{1}{2}f_{max} < f_{min} \leq \frac{4}{5}f_{max} \\ 0.7, & \frac{4}{5}f_{max} < f_{min} \leq f_{max} \end{cases} \quad (29)$$

$$P_m = \begin{cases} P_m^{min} - (0.05 - P_m^{min}) \left(\frac{\bar{f}}{f_{max}} - \frac{f - \bar{f}}{2(f_{max} - \bar{f})} \right), & \bar{f} \geq f \\ P_m^{min}, & \bar{f} < f \end{cases} \quad (30)$$

$$P_m^{min} = \begin{cases} 0.01, & f_{min} \leq \frac{1}{2}f_{max} \\ 0.02, & \frac{1}{2}f_{max} < f_{min} \leq \frac{4}{5}f_{max} \\ 0.03, & \frac{4}{5}f_{max} < f_{min} \leq f_{max} \end{cases} \quad (31)$$

3.4.4. Chromosome Repair Strategies

Since crossover and mutation operations in the machine assignment (MS) layer and transport resource assignment (RS) layer may produce locally infeasible or irrational assignments, a chromosome repair mechanism is introduced to restore feasibility following genetic operations. The repair principles are as follows:

1. Machine Assignment Correction: If a machine assigned at a given position in the MS layer does not belong to the set of valid candidate machines for the corresponding process, a new machine is randomly selected from the candidate set to replace the invalid assignment.
2. Transport Requirement Validation: If a process at a given position in the RS layer does not require transportation, the corresponding gene position is marked as invalid to reflect that no transport resource is needed.

3. Transport Robot Assignment: If a process at a given position in the RS layer requires transportation but lacks a valid transport robot ID, a valid robot ID is randomly selected from the set of available transport robots to fill the gap.

This repair strategy ensures that offspring generated after genetic operations remain interpretable by subsequent decoding procedures, thereby maintaining the stability and feasibility of the population evolution throughout the optimization process.

3.5. Variable Neighborhood Search

VNS is a classic local search metaheuristic that systematically changes neighborhood structures to escape local optima and explore high-quality solutions [26]. In this paper, VNS is integrated into ENSGA-II to enhance optimization of the FJSP-TRB problem.

3.5.1. Neighborhood Structures

1. N1: Critical Path Operation Rearrangement Strategy

Critical operations on the critical path are identified and selectively relocated to alleviate congestion in workshops with limited transport capacity. By moving these operations from congested segments to alternative paths, overall workshop congestion is reduced, and the arrival times of operations are adjusted, improving scheduling performance.

2. N2: Load-Release Strategy for Heavily Utilized Machines

During workshop operations, machine utilization may become imbalanced. The machine with the highest load is identified by evaluating the workload distribution across all machines. Some operations are then transferred from this machine to alternative available machines, thereby reducing resource idleness, minimizing energy consumption, and balancing the load across the workshop.

3. N3: Transport Equipment Allocation Adjustment Strategy

Transport capacity often limits production efficiency, and the allocation of transport robots directly affects workpiece arrival times and subsequent completion times. In this strategy, two processes are randomly selected and their assigned transport robots are reassigned to other available robots. The transport routes are subsequently replanned, improving waiting conditions, reducing travel time, and lowering energy consumption without modifying the operation sequence.

4. N4: Buffer Transfer Optimization Strategy

Utilizing a shared buffer zone allows early release of the current processing machine but introduces additional loading, transportation, and waiting operations. Improper buffer usage can increase total energy consumption and prolong transfer times between successive processes. For tasks with long buffer waiting times, the strategy involves: reselecting candidate machines for the next process to reduce buffer waiting time; reallocating subsequent transport robots to shorten transport delays after exiting the buffer; and, where feasible, adjusting transfer tasks to wait at the original machine before being transported directly to the target machine, thereby reducing transfer waiting time.

3.5.2. Procedures of Variable Neighborhood Search

Step 1: Initialization. Extract some elite individuals from the external archive set as the initial solution S , set the termination condition, and compute the objective function values.

Step 2: Perturbation operation. Perform a neighborhood perturbation on the initial solution to generate a perturbed solution.

Step 3: Feasibility repair and decoding. After perturbation, check the feasibility of the solution. If infeasible assignments occur in either the machine assignment layer or the transport robot assignment layer, perform a repair procedure. Then, decode the repaired chromosome using a decoding strategy to obtain the workpiece processing schedule,

machine occupancy schedule, and transport task schedule, and compute the corresponding multi-objective function values.

Step 4: Non-dominated relationship judgment and update of the optimal solution. If the new solution dominates the original solution, replace the original solution with the new one and store it in the external archive set. If the new solution and the original solution are non-dominated with respect to each other, retain the individual with better diversity based on crowding distance. If the new solution is dominated by the original solution, discard the new solution.

Step 5: Switch neighborhood structure. If the new solution is accepted, restart the search from the first neighborhood; otherwise, continue the search with the next neighborhood.

Step 6: Update of the External Archive. Compare all feasible new solutions obtained by variable neighborhood search with the existing solutions in the archive set one by one, remove dominated solutions, and retain non-dominated solutions. If the number of solutions in the archive set exceeds the preset capacity, delete solutions located in densely distributed regions of the objective space according to crowding distance, while ensuring both convergence and diversity of the Pareto solution set.

By integrating the above VNS procedure into ENSGA-II, the proposed algorithm can exploit high-quality local regions around elite solutions while preserving the global exploration ability of the evolutionary framework.

4. Experiment

4.1. Experimental Data

Since no suitable test data currently exists for the problem under study, the Brandimarte benchmark dataset was extended to generate 10 test instances, tailored to the problem's characteristics. Table 3 outlines the scale of these test instances, while Table 4 presents the transportation time schedule for MK01, where the LU zone is designated as the material storage area.

Table 3. Scale of test cases.

Numerical Example	Number of Workpieces	Number of Processing Machines	Number of Transport Robots
MK01	10	6	2
MK02	10	6	2
MK03	15	8	2
MK04	15	8	3
MK05	15	4	2
MK06	10	15	4
MK07	20	5	2
MK08	20	10	3
MK09	20	10	3
MK10	20	15	4

Table 4. Transportation Time.

Machine	M1	M2	M3	M4	M5	M6	LU
M1	0	1	3	1	2	1	1
M2	1	0	2	1	1	3	3
M3	3	2	0	1	3	2	1
M4	1	1	1	0	1	2	2
M5	2	1	3	1	0	1	2
M6	1	3	2	2	1	0	3
LU	1	3	1	2	2	3	0

4.2. Experimental Parameters

To determine the optimal combination of algorithmic parameters, an orthogonal experimental design was employed, focusing on three critical factors: population size (N), maximum number of iterations, and number of domain-changing searches. The levels for each factor were configured as follows: population size = {30, 60, 90, 120}, maximum number of iterations = {40, 60, 80, 100}, and number of domain-changing searches = {20, 30, 40, 50}. Based on these factor levels, an appropriate orthogonal array was constructed for the experiments. The MK01 test case was chosen for execution, with 20 independent runs of the algorithm performed for each parameter combination. Following each run, a set of non-dominated reference solutions was produced. The Inverted Generational Distance (IGD) metric was applied to assess these solutions, and the average IGD values are summarized in Table 5.

Table 5. Orthogonal Experimental Analysis.

Experiments	N	T	k	IGD
1	40	60	20	0.2796
2	40	80	30	0.3514
3	40	100	40	0.2050
4	40	120	50	0.3196
5	60	60	30	0.2823
6	60	80	20	0.2524
7	60	100	50	0.1817
8	60	120	40	0.2655
9	80	60	40	0.2054
10	80	80	50	0.2126
11	80	100	20	0.1839
12	80	120	30	0.1623
13	100	60	50	0.3493
14	100	80	40	0.2004
15	100	100	30	0.2230
16	100	120	20	0.3230

Based on the data in Table 5, Figure 13 summarizes the performance trends of the algorithm at each factor level.

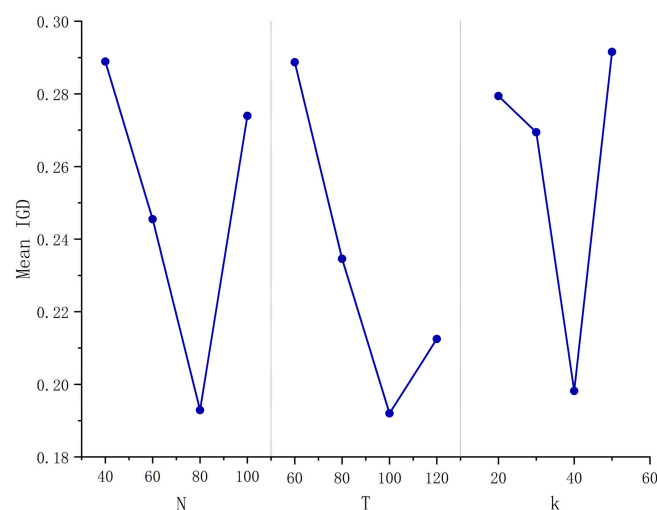


Figure 13. Trend of Parameter Levels.

The experimental results show that the optimal parameter combination for the algorithm is as follows: an initial population size of 80, a maximum of 100 iterations, and 40 domain-changing searches.

4.3. Performance Metrics

To evaluate the quality of the Pareto solution set generated by the proposed algorithm, two commonly used metrics are utilized: Inverted Generational Distance (IGD) and Hypervolume (HV). IGD is a measure for assessing the performance of multi-objective optimization algorithms, indicating how close the obtained solution set is to the true Pareto frontier. A smaller IGD value implies that the solution set is closer to the theoretical optimum. IGD is computed by calculating the distance between each solution in the true Pareto front and its nearest counterpart in the obtained set, and then averaging these distances. The IGD is formally defined as:

$$IGD = \frac{1}{L} \sum_{l=1}^L \min_{u \in \{1, 2, \dots, U\}} d_i(H_l, K_u) \quad (32)$$

$$d_i(H_{l,i}, K_u) = \sqrt{\sum_{tar}^{TAR} (h_{l,i,tar}, k_{l,i,tar})^2} \quad (33)$$

where L represents the population size, $H_{l,i}$ refers to the l th solution in the i th individual of the population solution set, U is the total number of solutions on the true Pareto frontier, and K_u denotes the u th solution on the true Pareto frontier. The term $d_i(H_{l,i}, K_u)$ denotes the distance metric between solution $H_{l,i}$ and solution K_u ; the Euclidean distance is adopted to quantify the distance between two solutions, as defined in the corresponding Equation (33).

Hypervolume (HV) is a commonly adopted indicator for assessing the overall quality of the non-dominated solution set produced by a multi-objective optimization algorithm. It captures both solution convergence toward the true Pareto frontier and the distributional diversity within the objective space. A larger HV value suggests that the solution set is closer to the true Pareto frontier and exhibits a more uniform distribution. The HV is computed as the measure of the union of hypercubes formed between each solution in the set S and a reference point r , formally defined as:

$$HV(S, r) = \lambda_3 \left(\bigcup_{y \in S} [y_1, r_1] \times [y_2, r_2] \times [y_3, r_3] \right) \quad (34)$$

where $\lambda_3(\cdot)$ denotes the Lebesgue measure in three-dimensional objective space, $y = (y_1, y_2, y_3) \in S$ is a solution in the obtained set, and $r = (r_1, r_2, r_3)$ is a predefined reference point dominated by all solutions in S .

4.4. Comparison Experiments

To verify the performance of FJSP-TRB and ENSGA-II, comparative experiments are conducted using MOPSO and NSGA-II. To guarantee experimental fairness, the extended Brandimarte benchmark instances are run ten times independently with identical parameter settings. The average values of the ten runs are adopted as the final evaluation metrics, and the corresponding experimental results are listed in Table 6.

Table 6. Optimization Objectives of Instance Data.

Experiments	ENSGA-II			NSGA-II			MOPSO		
	F1	F2	F3	F1	F2	F3	F1	F2	F3
1	182	12,558	52	240	15,680	60	184	12,970	63
2	179	11,606.5	7	210	12,823.4	11	186	11,706.7	10
3	862	56,281.5	70	965	65,838.4	83	939	62,512.6	73
4	357	23,998.4	40	370	25,196.3	46	373	43,996.2	39
5	555	33,069.8	10	740	54,036	52	558	34,810.1	26
6	536	43,094.3	68	537	43,365.6	74	816	67,525.2	120
7	590	41,365.4	87	642	50,622	96	610	47,836.3	89
8	1721	124,494.2	523	1670	125,534.1	524	1771	130,578.5	527
9	1547	112,438.8	124	1999	143,018.4	132	1642	122,424.3	98
10	1314	109,787.1	228	1316	112,767	229	1402	118,689.8	235

An analysis of the experimental results shows that, compared with NSGA-II, the proposed ENSGA-II algorithm performs better on all 10 test instances with respect to objective f_1 , and yields superior results for objectives f_2 and f_3 in most cases. These findings indicate that ENSGA-II achieves a clear overall improvement over the baseline NSGA-II algorithm on the WA01–WA10 instances. Similarly, when compared with MOPSO, ENSGA-II exhibits better performance for f_1 , f_2 , and f_3 in most of the 10 test instances. These results confirm the effectiveness of ENSGA-II in addressing the multi-objective flexible workshop scheduling problem with integrated transport and shared buffering, providing superior convergence, balanced machine utilization, and energy-efficient scheduling.

The IGD and HV values produced by the three algorithms were examined separately. Each of the 10 test instances was run independently multiple times, and the mean results were calculated to ensure statistical reliability. The corresponding performance metrics are reported in Table 7.

Table 7. Comparison of IGD and HV Values.

Experiments	IGD			HV		
	ENSGA-II	NSGA-II	MOPSO	ENSGA-II	NSGA-II	MOPSO
1	0.09214	0.10976	0.13649	0.03175	0.02187	0.01549
2	0.08734	0.21081	0.26145	0.02823	0.01639	0.02593
3	0.13056	0.18237	0.25368	0.04689	0.02349	0.01576
4	0.06538	0.12189	0.09372	0.18562	0.16719	0.01879
5	0.20318	0.20854	0.41693	0.21435	0.05238	0.08536
6	0.11539	0.22671	0.38924	0.09395	0.05493	0.07868
7	0.29045	0.33962	0.94617	0.28513	0.13347	0.17516
8	0.26157	0.40319	0.68015	0.13347	0.09736	0.08237
9	0.26683	0.29641	0.73126	0.19036	0.05736	0.00942
10	0.17352	0.22968	0.30247	0.06424	0.02147	0.0069

Across all 10 test instances, the ENSGA-II algorithm consistently achieved more favorable IGD and HV values than NSGA-II and MOPSO, indicating that ENSGA-II exhibits a clear advantage in solving multi-objective problems in terms of both convergence toward the true Pareto front and uniformity of solution distribution. In terms of overall algorithm performance, NSGA-II and MOPSO demonstrated comparable results, whereas ENSGA-II achieved significantly better outcomes than NSGA-II. These findings offer preliminary evidence supporting the effectiveness of the proposed improvement strategies in ENSGA-II. Overall, the experimental results confirm that ENSGA-II is effective for solving integrated scheduling problems with shared buffers and transport resources, delivering improved solution quality and enhanced system efficiency.

4.5. Experimental Validation

In addition to evaluating the algorithm on the benchmark dataset, relevant case studies were conducted to further validate the proposed model and assess the performance of ENSGA-II in the transportation-integrated multi-objective flexible scheduling problem. The specific machining parameters for these case studies are presented in Table 8.

Table 8. Machining Data.

Job	Process	Machine Processing Time of Operation				
		M1	M2	M3	M4	M5
J1	O _{1,1}	—	10	—	16	12
	O _{1,2}	9	—	14	—	—
	O _{1,3}	—	13	17	10	—
	O _{2,1}	—	16	—	13	—
	O _{2,2}	—	15	—	10	17
J2	O _{2,3}	16	—	13	—	—
	O _{2,4}	9	—	—	—	12
	O _{2,5}	—	—	15	11	17
	O _{2,6}	—	12	13	16	—
	O _{3,1}	—	—	—	14	17
J3	O _{3,2}	12	15	—	—	14
	O _{3,3}	—	11	19	—	—
	O _{3,4}	10	—	—	8	—
J4	O _{4,1}	—	—	14	—	16
	O _{4,2}	—	9	—	14	16
	O _{5,1}	15	—	12	—	10
J5	O _{5,2}	—	16	—	—	9
	O _{5,3}	11	15	—	9	—
	O _{5,4}	—	—	10	—	—
	O _{5,5}	—	—	8	12	15
	O _{6,1}	8	—	10	—	—
J6	O _{6,2}	—	11	—	16	15
	O _{6,3}	9	12	—	—	15
	O _{6,4}	—	—	—	13	15
	O _{6,5}	—	14	10	12	—
	O _{7,1}	9	—	—	14	—
J7	O _{7,2}	—	11	—	—	17
	O _{7,3}	9	—	13	15	16
	O _{7,4}	—	9	—	13	—

In the presented case study, seven job types are processed across five machine tools; transport robots are responsible for workpiece handling as well as loading and unloading operations between machines. The table presents the transport times between machines, where “LU” refers to the loading/unloading area and the shared buffer zone. At time zero, all workpieces are positioned in the LU zone. “Home” represents the initial position of the transport robots, with all robots located at the Home zone at time zero, ready to begin the transportation tasks. Table 9 shows the transportation time schedule.

Table 9. Transfer Time.

Machine	M1	M2	M3	M4	M5	LU	Home
M1	0	2	3	1	1	2	4
M2	2	0	1	1	3	4	2
M3	3	1	0	2	4	5	1
M4	1	1	2	0	2	3	3
M5	1	3	4	2	0	1	5
LU	2	4	5	3	1	0	6
Home	4	2	1	3	5	6	0

Transport robots are employed to transfer workpieces and carry out loading and unloading operations between machine tools. The transport robot's rated power under load is 3 kW, its no-load power is 1.5 kW, its standby power is 0.5 kW, and its operating power is 0.8 kW. The relevant power specifications for each machine tool are shown in Table 10.

Table 10. Rated power and standby power of each machine.

Machine	M1	M2	M3	M4	M5
Machine processing power	13	9	6	17	10
Machine idle power	3.45	1.25	1.03	3.81	2.75

The above case study was implemented and analyzed using MATLAB 2022b. Figure 14 shows the Pareto front obtained from multiple independent runs of the ENSGA-II algorithm, illustrating the distribution of Pareto-optimal solutions in the three-objective space of makespan, total energy consumption, and machine load deviation.

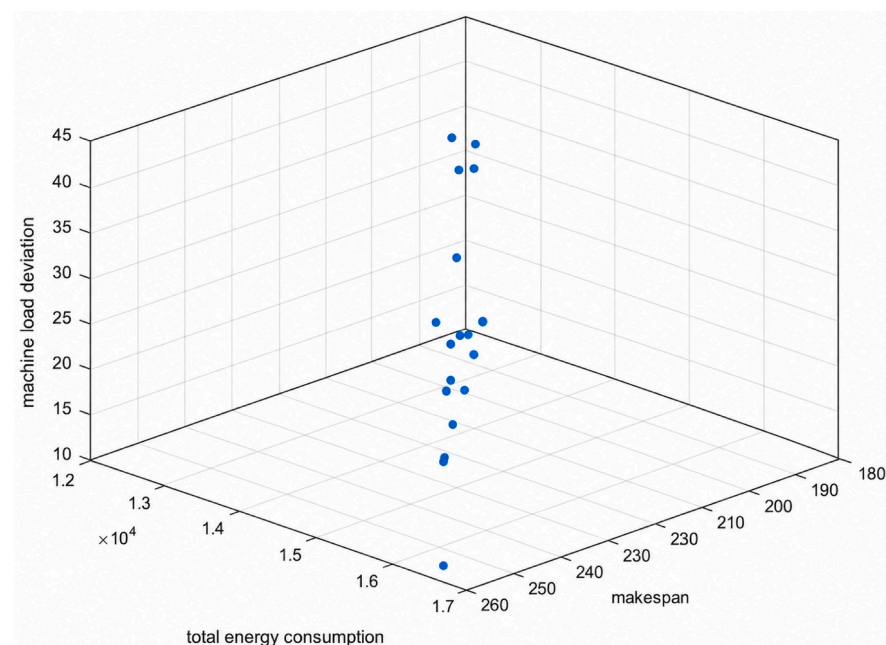


Figure 14. 3D scatter plot of the scheduling scheme.

Figure 14 illustrates the trade-off relationships among the three optimization objectives, namely makespan, total energy consumption, and machine load range. It can be observed that these objectives cannot be minimized simultaneously by a single solution. Solutions with a smaller makespan usually require more intensive use of machines and more frequent dispatching of transport robots, which may increase both machine idle energy and robot transportation energy. Conversely, energy-oriented solutions tend to reduce unnecessary loaded and empty robot movements as well as redundant buffer transfers, but this may prolong the waiting time of some operations and lead to a larger makespan.

The trade-off between makespan and machine load range is also evident. When the algorithm gives priority to reducing the maximum completion time, operations are more likely to be assigned to machines with shorter processing times or earlier availability. This may cause workload concentration on several key machines and increase the machine load range. In contrast, reducing the machine load range requires transferring some operations to less-loaded machines, which improves workload balance but may sacrifice processing efficiency and increase the completion time.

In addition, the shared-buffer mechanism further strengthens the coupling among the three objectives. Sending a workpiece to the shared buffer can release the current machine earlier and reduce machine blocking, but it may also introduce additional loading, transportation, unloading, and waiting operations. Therefore, the use of the shared buffer should be determined according to the joint effects on production efficiency, transportation energy consumption, and machine load balance. The Pareto solutions shown in Figure 14 provide different scheduling alternatives for decision-makers under different production preferences, such as time-oriented, energy-oriented, and balanced scheduling strategies.

Figure 15 illustrates how the three objectives—completion time, total energy consumption, and machine load deviation—vary with the number of algorithm iterations, thereby revealing the convergence process and the gradual improvement in solution quality over successive iterations. As illustrated in Figure 15, with increasing iterations, all scheduling objectives exhibit continuous improvement, demonstrating strong convergence. In other words, as the population evolves, the three optimization objectives—completion time, total energy consumption, and machine load deviation—progressively improve.

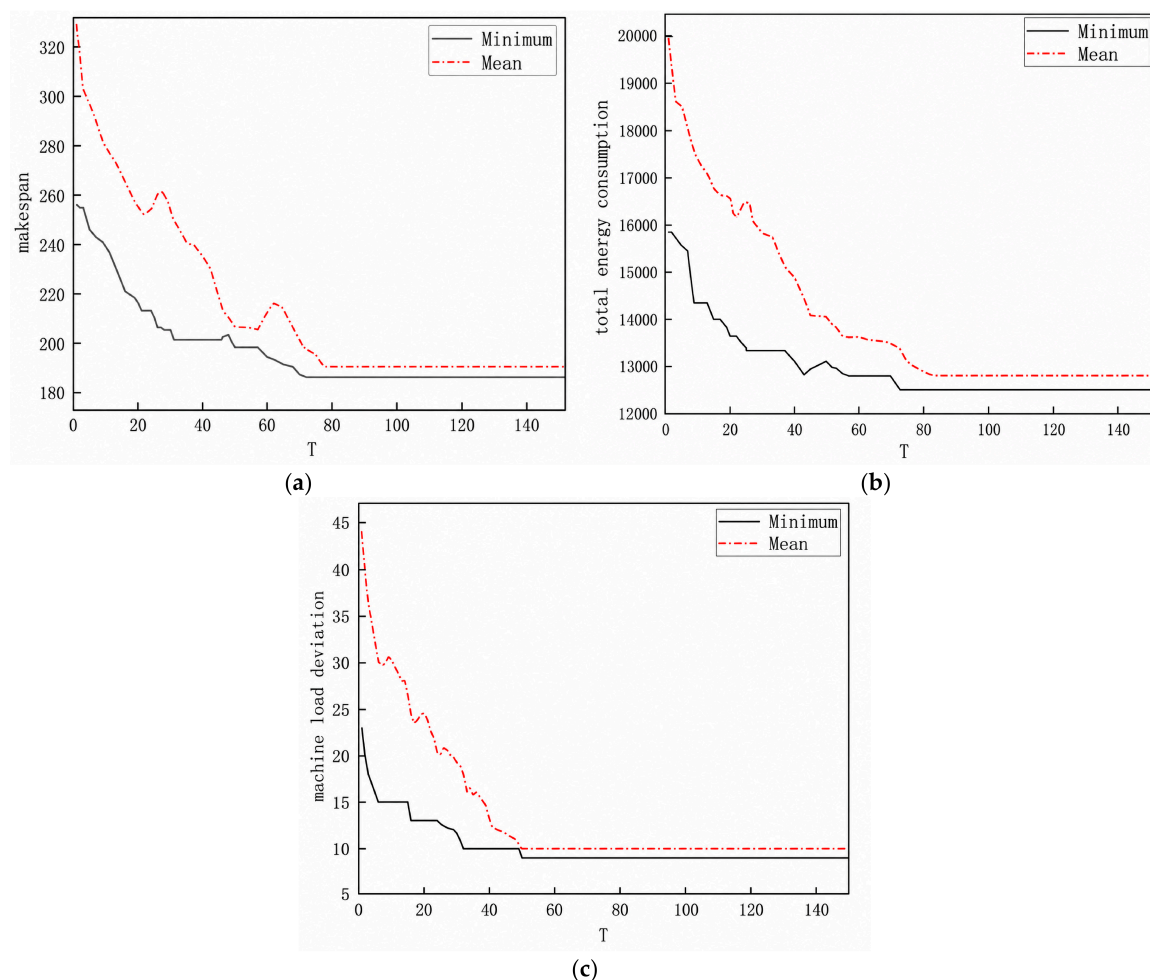


Figure 15. Level Convergence Curve. (a) Makespan; (b) total energy consumption; (c) machine load deviation.

Figure 16 presents the Gantt chart generated by ENSGA-II for this case study. The horizontal axis indicates the time required for processing, whereas the vertical axis represents the machine indices. Each workpiece is assigned a distinct color, and the length of each rectangular bar indicates the processing time of the corresponding operation. For example, J2-1 denotes that the first operation of Workpiece 2 is processed on Machine M4, with a

maximum completion time of 190, a maximum machine load of 12,878, and a machine load range of 10. This visualization effectively illustrates the scheduling sequence, machine assignment, and operation durations, providing an intuitive assessment of the algorithm's ability to optimize integrated workshop scheduling.

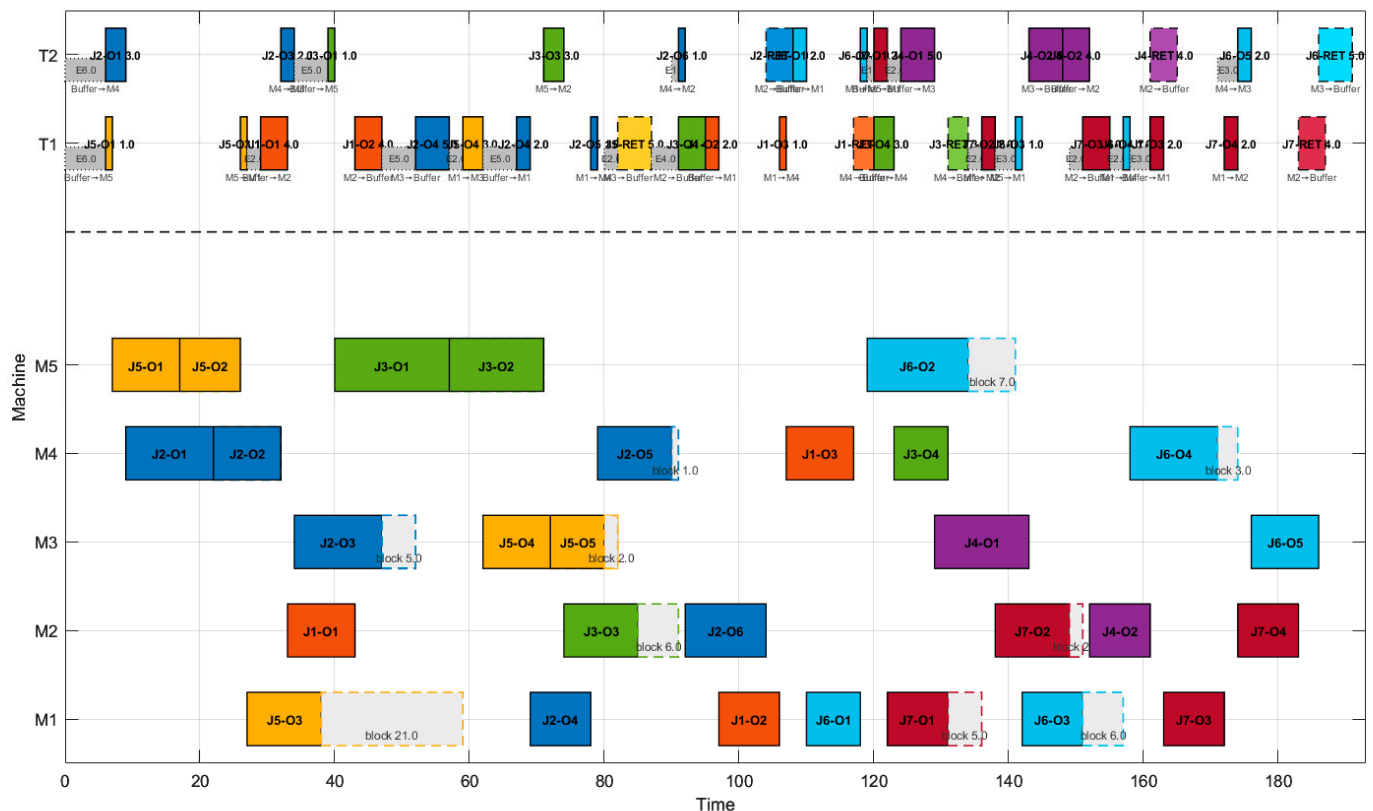


Figure 16. Gantt chart of the scheduling scheme.

5. Conclusions

This paper examines the scheduling problem in flexible workshops where industrial robots are responsible for inter-process transportation. To address this issue, a flexible workshop scheduling model with shared buffers is established, with the objectives of minimizing the maximum completion time, total energy consumption in both processing and transportation, and the variation in machine loads. The ENSGA-II algorithm tailored to this model is proposed, and simulation experiments demonstrate its superior performance in solving the FJSP-TRB problem. The main contributions of this study are summarized as follows:

1. **Innovation in Problem Modeling.** This study establishes a multi-objective, energy-aware transport-integrated flexible job shop scheduling model with shared buffer constraints, which minimizes makespan, total energy consumption of processing and transport machines, and machine load range. The model explicitly accounts for the energy consumption of transport robots across all operating states, including empty travel, loaded travel, and idle operation.
2. **Innovation in Decoding Strategy.** For the FJSP-TRB, a time-window-based path planning decoding strategy that accounts for both buffer constraints and transport conflicts is designed. This strategy resolves conflicts arising from buffer constraints and the process of workpiece transfer by transport robots during scheduling. The proposed decoding strategy can effectively plan the transportation paths of transport robots, reduce waiting time, and improve processing efficiency.

3. **Innovations in Algorithm Design.** The ENSGA-II algorithm is proposed to solve the problem. A time-window-based path planning decoding strategy is introduced to address the characteristics of shared buffers and transport robots, accounting for scenarios involving buffer constraints and transport conflicts. Furthermore, a variable-domain search mechanism is incorporated to perform local search operations on non-dominated solutions in the external archive, ensuring both solution diversity and convergence.
4. **Application innovation.** Traditional FJSP or FJSPT mainly focus on machine assignment and operation sequencing, or use only dedicated transport resources. In contrast, this paper simultaneously considers industrial robots performing both transportation and loading/unloading tasks, and coordinates multiple resources in a shared buffer environment. This design makes the model more consistent with the operational logic of actual smart manufacturing workshops while supporting multi-objective optimization.

Although this paper makes several contributions, it also has some limitations that simultaneously point to promising future research directions.

1. **Limitations of the model.** The current model assumes a static and predictable production environment, i.e., all machines and robots are available at time zero, no equipment failures occur, and no rush orders are inserted. However, in real flexible manufacturing systems, unexpected disturbances such as equipment failures, tool wear, and rush order insertions occur frequently. These stochastic events may invalidate pre-computed schedules and significantly degrade system performance.

2. **Limitations of robot operating conditions.** The model treats transport robots as idealized agents, assuming constant speed regardless of the load and ignoring battery charging cycles as well as scheduled equipment maintenance downtime. In practice, the speed of a loaded robot is generally lower than that of an unloaded robot, which directly affects transport durations and the feasibility of time-window-based path planning. Moreover, neglecting charging and maintenance implies unlimited robot availability, which is unrealistic for long-term or high-intensity operations. These simplifications may lead to underestimation of total transport time and overestimation of the effectiveness of conflict resolution strategies.

3. **Constraints induced by shared buffer capacity under high-load stochastic logistics scenarios.** Although the model introduces a shared buffer to mitigate machine blocking, it assumes that the buffer has sufficient physical space and that workpieces always follow the FIFO rule, without considering buffer capacity saturation. In high-load or stochastic arrival scenarios, the shared buffer itself may become a new bottleneck. Excessive queue lengths increase waiting times, complicate rescheduling decisions, and may even cause deadlocks when transport robots are unavailable. The current model does not explicitly address buffer capacity constraints. This shortcoming is particularly prominent for shops with limited space or for just-in-time production.

4. **Regarding the limitation of the problem scale.** The case studies used for validation in this research are relatively small in scale. Therefore, they are insufficient to fully demonstrate the applicability of the proposed method to larger-scale industrial production scenarios or to verify its scalability.

5. **Research extension directions.** To overcome the above limitations, the following extended studies are proposed:

Construct a stochastic scheduling model: Incorporate random disturbances into the multi-objective optimization framework.

Develop a real-time rescheduling strategy: Design a reactive rescheduling mechanism that can quickly adjust the original schedule when disturbances occur. A rolling horizon

method can be combined with the proposed ENSGA-II algorithm as a fast re-planner, striking a balance between solution quality and computational response speed. Event-triggered rescheduling approaches may also be explored.

Design an adaptive ENSGA-II algorithm: Enhance the current ENSGA-II by introducing adaptive mechanisms, such as online parameter adjustment based on population diversity or machine learning for dynamic neighborhood search operator selection. This will improve the algorithm's robustness under different problem scales and uncertainty levels.

Model shared buffer capacity: Extend the FJSP-TRB model to explicitly include limited buffer capacity and dynamic buffer allocation rules. The decoding scheme should treat buffer occupancy as a constraint, and new neighborhood moves should be introduced to prevent or resolve buffer-induced deadlocks.

Expanding the research scale: As the problem scale increases, the number of operations, alternative machines, transportation tasks, and robot time windows will also increase accordingly. This may lead to longer decoding times and higher overall computational costs. For large-scale industrial scenarios, the scalability of the proposed algorithm can be improved through the following strategies. First, the fitness evaluation and decoding processes of different individuals within the same population generation are independent of each other; therefore, parallel computing can be adopted to accelerate the solution process. Second, efficient data structures can be used to store and update the time-window information of machines and robots, thereby reducing repeated conflict-checking operations. Third, the execution frequency of variable neighborhood search can be adaptively adjusted to avoid excessive ineffective local searches when the population size becomes large. Fourth, a large workshop can be divided into multiple production areas. The proposed algorithm can first be applied to complete internal scheduling within each area, and then inter-area transportation tasks can be coordinated in a unified manner.

By addressing the above limitations in future work, the proposed integrated scheduling framework can be better applied to real-world smart manufacturing environments, where uncertainties, energy constraints, and logistics congestion are the norm rather than the exception.

Author Contributions: Conceptualization, X.L., L.Z. and Y.W.; methodology, X.L., L.Z. and Y.W.; software, X.L.; validation, X.L.; investigation, X.L., H.L. and H.Z.; resources, H.L. and H.Z.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L., L.Z. and Y.W.; supervision, Y.W., H.L., H.Z. and L.Z.; project administration, Y.W., H.L., H.Z. and L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Education of China (Grant No. LJ212411258030) and the Key R&D Joint Program of Liaoning Provincial Science and Technology Program (Grant No. 2025JH2/101800357).

Data Availability Statement: Data are contained within the article.

Acknowledgments: I would like to express my sincere gratitude to my university for providing excellent research facilities and a supportive academic environment throughout my study. I am especially grateful to Yuan-Gang Wang, for their invaluable guidance, encouragement, and insightful suggestions during the course of this research. Their expertise and patience have been essential to the completion of this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Fan, J. A Study of Job-Shop Scheduling Based on Multi-Resource Constrained. *Ind. Eng. J.* **2017**, *20*, 44–50.
2. Angelopoulos, J.; Alexopoulos, K. The Role of Digital Transformation in Manufacturing under the Light of AI Integration. In *Artificial Intelligence, Data and Robotics: Foundations, Transformations and Future Directions*; Springer: Cham, Switzerland, 2026; pp. 201–227.
3. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [\[CrossRef\]](#)
4. Zhou, J.; Li, P.; Zhou, Y.; Wang, B.; Zang, J.; Meng, L. Toward New-Generation Intelligent Manufacturing. *Engineering* **2018**, *4*, 11–20. [\[CrossRef\]](#)
5. Liang, T.; Zhou, L.; Jiang, Z. Integrated Scheduling of Production and Material Delivery for the Intelligent Manufacturing System. *Int. J. Prod. Res.* **2025**, *63*, 882–903.
6. Jamrus, T.; Chien, C.-F.; Gen, M.; Sethanan, K. Hybrid Particle Swarm Optimization Combined with Genetic Operators for Flexible Job-Shop Scheduling under Uncertain Processing Time for Semiconductor Manufacturing. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 32–41. [\[CrossRef\]](#)
7. Lu, C.; Li, X.; Gao, L.; Liao, W.; Yi, J. An Effective Multi-Objective Discrete Virus Optimization Algorithm for Flexible Job-Shop Scheduling Problem with Controllable Processing Times. *Comput. Ind. Eng.* **2017**, *104*, 156–174.
8. Liu, R.; Jia, Q.; Yu, H.; Gao, K.; Fu, Y.; Yin, L. Bi-Objective Integrated Scheduling of Job Shop Problems and Material Handling Robots with Setup Time. *Mathematics* **2025**, *13*, 447. [\[CrossRef\]](#)
9. Berterottière, L.; Dauzère-Pérès, S.; Yugma, C. Flexible Job-Shop Scheduling with Transportation Resources. *Eur. J. Oper. Res.* **2024**, *312*, 890–909.
10. Luan, F.; Zhao, H.; Liu, S.Q.; He, Y.; Tang, B. Enhanced NSGA-II for Multi-Objective Energy-Saving Flexible Job Shop Scheduling. *Sustain. Comput. Inform. Syst.* **2023**, *39*, 100901.
11. Sun, J.; Zhang, G.; Lu, J.; Zhang, W. A Hybrid Many-Objective Evolutionary Algorithm for Flexible Job-Shop Scheduling Problem with Transportation and Setup Times. *Comput. Oper. Res.* **2021**, *132*, 105263.
12. Meng, L.; Cheng, W.; Zhang, B.; Zou, W.; Fang, W.; Duan, P. An Improved Genetic Algorithm for Solving the Multi-AGV Flexible Job Shop Scheduling Problem. *Sensors* **2023**, *23*, 3815. [\[CrossRef\]](#)
13. Zhang, G.; Yan, S.; Song, X.; Zhang, D.; Guo, S. Evolutionary Algorithm Incorporating Reinforcement Learning for Energy-Conscious Flexible Job-Shop Scheduling Problem with Transportation and Setup Times. *Eng. Appl. Artif. Intell.* **2024**, *133*, 107974. [\[CrossRef\]](#)
14. Xu, Z.; Jia, Q.; Gao, K.; Fu, Y.; Yin, L.; Sun, Q. Integrated Scheduling of Multi-Objective Job Shops and Material Handling Robots with Reinforcement Learning Guided Meta-Heuristics. *Mathematics* **2024**, *13*, 102. [\[CrossRef\]](#)
15. Zhang, H.; Qin, C.; Zhang, W.; Xu, Z.; Xu, G.; Gao, Z. Energy-Saving Scheduling for Flexible Job Shop Problem with AGV Transportation Considering Emergencies. *Systems* **2023**, *11*, 103. [\[CrossRef\]](#)
16. Kasapidis, G.A.; Paraskevopoulos, D.C.; Mourtos, I.; Repoussis, P.P. A Unified Solution Framework for Flexible Job Shop Scheduling Problems with Multiple Resource Constraints. *Eur. J. Oper. Res.* **2025**, *320*, 479–495. [\[CrossRef\]](#)
17. Zhang, B.; Zuo, T.; Wang, Z. Improved Salp Swarm Algorithm for Multi-Resource Job Shop Scheduling Problem with Buffer Capacity Constraints. *Soft Comput.* **2025**, *30*, 635–652. [\[CrossRef\]](#)
18. Inoue, K.; Aoyama, H. Scheduling for Assembly Line with Human–Robot Collaboration. *J. Korean Soc. Precis. Eng.* **2023**, *40*, 163–173. [\[CrossRef\]](#)
19. Aschauer, A.; Roetzer, F.; Steinboeck, A.; Kugi, A. An Efficient Algorithm for Scheduling a Flexible Job Shop with Blocking and No-Wait Constraints. *IFAC-PapersOnLine* **2017**, *50*, 12490–12495. [\[CrossRef\]](#)
20. Sun, X.; Shen, W.; Fan, J.; Vogel-Heuser, B.; Bi, F.; Zhang, C. Deep Reinforcement Learning-Based Multi-Objective Scheduling for Distributed Heterogeneous Hybrid Flow Shops with Blocking Constraints. *Engineering* **2025**, *46*, 278–291. [\[CrossRef\]](#)
21. Wen, T.; Guan, T. Hybrid Flow Shop Scheduling with Limited Buffers Considering Energy Consumption and Transportation. *J. Syst. Simul.* **2024**, *36*, 1344–1358.
22. Shakeri, Z.; Benfriha, K.; Varmazyar, M.; Talhi, E.; Quenehen, A. Production Scheduling with Multi-Robot Task Allocation in a Real Industry 4.0 Setting. *Sci. Rep.* **2025**, *15*, 1795. [\[CrossRef\]](#)
23. Haj Mouldi, A.B.; Nouri, M. Integrated Genetic Algorithm with Dispatching Rules to Solve the Flexible Job Shop Scheduling Problem under Multi-AMR Transportation Constraints. *IFAC-PapersOnLine* **2024**, *58*, 652–657. [\[CrossRef\]](#)
24. Dong, X.; Wan, G.; Zeng, P. A Heuristic-Assisted Deep Reinforcement Learning Algorithm for Flexible Job Shop Scheduling with Transport Constraints. *Complex Intell. Syst.* **2025**, *11*, 210. [\[CrossRef\]](#)

25. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
26. Mladenović, N.; Hansen, P. Variable Neighborhood Search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.