Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi-Objective Distributed Flexible Job Shop Schedulingwith Transfer Based on Critical Path Graph Guided DRL

Longyan Xu
Hubei University of Automotive Technology
Li Li
Hubei University of Automotive Technology
Kai Fang
Hubei University of Automotive Technology
Jinfeng Ye
Hubei University of Automotive Technology
NengHui Huang
xuly_dy@huat.edu.cn

Science and Technology College of Hubei University of Arts and Science

Research Article

Keywords: Distributed flexible job shop scheduling, Multi-objective optimization, Deep reinforcement learning, Graph convolutional networks, Green manufacturing

Posted Date: July 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9350585/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Multi-Objective Distributed Flexible Job Shop Scheduling with Transfer Based on Critical Path Graph Guided DRL

Longyan Xu[1,2], Li Li[1,2], Kai Fang[1,2], Jinfeng Ye[1,2], NengHui Huang[3*]

[1]School of Robotics and Automation, Hubei University of Automotive Technology, Shiyan, 442002, China.
[2] Shiyan Key Laboratory of Intelligent Equipment and Smart Operation & Maintenance for Automobile Manufacturing, Hubei University of Automotive Technology, Shiyan, 442002, China.
[3]School of Mechanical and Automotive Engineering, Science and Technology College of Hubei University of Arts and Science, Xiangyang, 441025, Hubei, China.

*Corresponding author(s). E-mail(s): xuly_dy@huat.edu.cn;
Contributing authors: 20060018@huat.edu.cn; 2297761221@qq.com;
20030015@huat.edu.cn; 20140016@huat.edu.cn;

Abstract

With the deepening of global production division and the promotion of green manufacturing policy, the multi-objective distributed flexible job shop scheduling problem (MO-DFJSP), which optimizes maximum completion time and total energy consumption simultaneously, has become an important direction of manufacturing system research. However, the existing solution methods based on evolutionary algorithms have two systemic shortcomings, the gene operator lacks the ability to perceive the scheduling structure, and the cross-generational population stagnation problem lacks a hierarchical response mechanism based on multidimensional indicators. In view of these limitations, this paper proposes a hybrid NSGA-II algorithm (HENSGA-DRL) based on disjunctive graph enhanced deep reinforcement learning. The core components include a critical-path graph-guided deep reinforcement learning operator (CGDRL) and a multi-metric state-aware hierarchical intervention strategy (MSHI), which act on intra-generational gene manipulation and cross-generational state regulation

respectively and cooperatively drive algorithm search. On the self-constructed DLA benchmark dataset, experimental results show that HENSGA-DRL is significantly superior to comparison algorithms in both convergence and diversity dimensions.

Keywords: Distributed flexible job shop scheduling, Multi-objective optimization, Deep reinforcement learning, Graph convolutional networks, Green manufacturing

1 Introduction

Job Shop Scheduling Problem (JSP) is the core link that connects production planning and actual execution in modern manufacturing systems. Its decision quality directly affects the production cycle, resource utilization rate, and operation cost of enterprises. The Flexible Job Shop Scheduling Problem (FJSP) introduces machine flexibility on the basis of the classical JSP, making machine allocation and operation sequencing highly coupled sub-decisions, and has been proved to be NP-hard [1].

With the continuous evolution of problem scenarios, two kinds of realistic pressures have pushed the research boundary further. The first is the demand for green manufacturing. The energy crisis and carbon emission reduction policies have prompted manufacturing enterprises to incorporate energy consumption into their scheduling objectives, and single-objective models that only minimize makespan can no longer reflect actual decision-making needs [2]. There is an essential conflict between makespan and total energy consumption: compressing the production cycle requires increasing the machine load rate, while reducing energy consumption tends to reduce the use of high-power machines, and the two objectives restrict each other in machine allocation decisions [3]. The second is the trend of distributed manufacturing. The global division of production labour has made the transfer of workpieces between geographically dispersed workshops a key factor affecting scheduling quality [4], and the coupling effect of heterogeneous machine capabilities across different factories and cross-factory transfer further aggravates the complexity of the problem [5]. These two pressures give rise to the Multi-objective Distributed Flexible Job Shop Scheduling Problem (MO-DFJSP), which simultaneously optimizes makespan and total energy consumption.

Evolutionary algorithms represented by genetic algorithms have become the mainstream framework for solving such problems [6], but the existing methods have systemic deficiencies at the following two levels. At the level of intra-generational gene manipulation, mutation operators have no ability to sense the scheduling structure, a large number of function evaluations are consumed on random perturbations of non-bottleneck operations, and the convergence efficiency of the Pareto front is fundamentally restricted. The existing work integrating reinforcement learning with evolutionary algorithms only uses reinforcement learning for hyperparameter regulation, and the perceptual ability stops at the parameter level without penetrating into the gene manipulation itself [7]. At the level of cross-generational state regulation,

population stagnation is another common bottleneck for evolutionary algorithms solving MO-DFJSP. The existing intervention mechanisms mostly rely on fixed threshold rules and lack the hierarchical response ability based on comprehensive evaluation of multi-dimensional indicators [8]. The limitations of these two levels point to the same core contradiction: how to simultaneously guarantee structural perception of the search direction and continuous vitality of the population evolution trajectory under limited computing resources.

Aiming at this contradiction, this paper proposes a hybrid NSGA-II algorithm based on disjunctive graph enhanced Deep Reinforcement Learning (DRL), namely HENSGA-DRL. The algorithm takes NSGA-II as the evolutionary backbone, designs the critical-path graph-guided DRL operator (CGDRL), constructs the constraint topology of the current scheduling solution through disjunctive graphs, extracts operation-level bottleneck features by graph convolutional networks, and then directly generates machine reassignment decisions via a PPO policy network and writes them back to chromosomes, so that each generation of mutation operations has the ability to continuously perceive and intervene on the global scheduling structure. At the same time, this paper introduces the multi-metric state-aware hierarchical intervention strategy (MSHI) to quantify population health status in real time with hypervolume change rate and objective-space diversity as monitoring indicators, and activates three-tier intervention measures according to the stagnation severity gradient to continuously maintain the health of the population evolution trajectory on the cross-generational time scale. The two modules act on intra-generational and cross-generational time scales respectively, collaboratively driving the algorithm to efficiently approximate a high-quality Pareto front in the complex search space of MO-DFJSP.

The main contributions of this paper are as follows.

(1) A complete multi-objective mathematical model covering processing energy consumption, idle energy consumption, and transportation energy consumption is established for MO-DFJSP, providing a unified formal description basis for algorithm design.

(2) The CGDRL operator is designed, embedding disjunctive graph structure perception and the PPO policy network into the evolutionary framework in the form of a gene operator, so that the mutation operation itself has directional decision-making capability for scheduling topology bottlenecks.

(3) The MSHI strategy is proposed to address the cross-generational population stagnation problem with a hierarchical intervention mechanism driven by multi-dimensional indicators, which forms intra-generational and cross-generational synergy with CGDRL on the time scale, overall improving the optimization efficiency of the algorithm in high-dimensional search space.

The rest of this paper is organized as follows. Section 2 summarizes related research work. Section 3 establishes the mathematical model of MO-DFJSP. Section 4 introduces the HENSGA-DRL algorithm in detail. Section 5 presents experimental results and analysis. Section 6 gives the conclusion.

2 Related Work

2.1 Research Status of Distributed Flexible Job Shop Scheduling

DFJSP is a natural extension of FJSP to multi-factory scenarios, and its research motivation stems from the fundamental contradiction that the single-factory production mode is unable to meet the demands of large-scale orders in real production.

Early research generally limited all operations of the same job to be completed in a single workshop. Under this constraint, scheduling decisions only need to handle machine allocation and operation sequencing within the factory, and the problem scale is relatively controllable. Meng et al. [9] established mixed integer linear programming and constraint programming models of DFJSP and systematically verified the optimality of multi-class benchmark instances. Zhang et al. [10] established an FJSP model with multiple time constraints and designed an improved genetic algorithm. Sun et al. [11] proposed a hybrid many-objective evolutionary algorithm for multi-objective FJSP with transportation and setup times. Lin et al. [12] further studied the distributed flexible job shop scheduling problem integrating process planning and shop scheduling and gave an effective solution method. The work in this period accumulated important achievements in establishing the basic model and algorithm framework, but forcibly binding operations to a single factory avoids the core contradiction of cross-factory collaboration, and when a factory encounters a capacity bottleneck, the scheduling solution cannot transfer load to other workshops, severely restricting overall production efficiency.

This unbalanced capacity utilization problem pushed subsequent research to introduce inter-factory workpiece transfer mechanisms, allowing different operations of a job to be completed in different workshops so that scheduling solutions can actively balance factory loads, but at the same time introducing new modeling challenges regarding the impact of transfer time on operation precedence constraints. Xie et al. [13] proposed a hybrid genetic tabu search algorithm for DFJSP with transportation. Li et al. [14] designed an improved grey wolf optimization algorithm for similar problems. Cao et al. [15] introduced an inverse model and adaptive neighborhood search cooperative optimizer, and their results show that the cross-factory transfer strategy can effectively relieve capacity bottlenecks and improve overall completion efficiency. Luo et al. [16] designed an improved memetic algorithm for DFJSP with worker constraints, further expanding the practical constraint dimension of the problem. However, the optimization objectives of the above work still focus on timeliness indicators such as completion time, lacking explicit modeling of the energy consumption generated during machine operation, a limitation that becomes increasingly apparent under the background of green manufacturing.

With the enforcement of sustainable manufacturing policies, energy consumption targets began to be systematically incorporated into the optimization framework of distributed scheduling. Zhang et al. [17] designed a multi-dimensional probabilistic model-driven evolutionary algorithm specifically for energy consumption efficiency. Gu et al. [18] proposed an improved memetic algorithm considering both transfer time

and machine start-stop constraints, showing that start-stop constraints have a non-negligible impact on energy consumption targets. Li et al. [19] proposed an adaptive memetic algorithm based on surprise popularity to achieve energy-oriented distributed scheduling optimization while maintaining population diversity. Du et al. [20] designed a hybrid estimation of distribution algorithm for distributed assembly FJSP. Wang et al. [21] combined Q-learning with artificial bee colony algorithm for distributed assembly scheduling with factory eligibility and transportation capacity constraints. Tian et al. [22] proposed a critical path-based variable neighborhood genetic algorithm for distributed assembly job shop scheduling. These works enable the conflict relationship between completion time and energy consumption to be explicitly quantified.

Throughout the above literature, whether in early single-factory assumption work or subsequent research introducing cross-factory transportation and energy consumption objectives, the evolutionary mutation operators adopted always remain at the level of random chromosome perturbation, with no perception of the distribution of critical paths and the position of bottleneck operations in the current scheduling solution. This means the algorithm cannot actively focus search pressure on the decision dimension with the greatest improvement potential, and the limited function evaluation budget is largely consumed on invalid perturbation of non-bottleneck operations. Therefore, this paper further explores a directed mutation mechanism with scheduling structure awareness under the MO-DFJSP framework.

2.2 Research Status of Multi-Objective Evolutionary Algorithms

The methods for solving multi-objective FJSP are usually divided into two categories: weighted methods and Pareto methods. Although the weighting method is simple to implement, it easily loses the non-convex part of the Pareto front when objective conflicts are large, and it is difficult to provide the complete trade-off information required by decision makers. The Pareto method approximates multiple objectives simultaneously by maintaining a non-dominated solution set without preset weights. He et al. [23] systematically accelerated the convergence process of large-scale multi-objective optimization from the perspective of problem reformulation, laying an important foundation for the theoretical development of the Pareto method. Chen et al. [24] combined fast non-dominated sorting with crowding distance to propose a self-learning genetic algorithm with NSGA-II as the backbone, achieving a good balance between convergence speed and population diversity maintenance, and has become one of the most widely used evolutionary paradigms for solving MO-DFJSP.

On this basis, researchers continue to explore ways to improve the solution efficiency of evolutionary algorithms. Pan et al. [25] designed a dual-population evolutionary algorithm based on feedback mechanism for energy-efficient optimization of fuzzy FJSP. Yu et al. [26] combined Q-learning with multi-objective meta-heuristics to achieve energy-efficient scheduling of distributed assembly permutation flowshops. Li et al. [27] fused knowledge-based reinforcement learning with estimation of distribution algorithm to guide the population evolutionary direction through historical experience, significantly improving the multi-objective solution quality of FJSP. Li et

al. [28] further proposed a learning memetic algorithm combining energy consumption optimization under type-2 fuzzy processing times with evolutionary search. These works reveal a common pattern: when the learning capability of reinforcement learning is used to influence the search behavior of evolutionary algorithms, algorithm performance can be systematically improved.

However, the role of reinforcement learning in the above improvement work always stays at parameter adjustment or operator selection, and the mutation operator itself remains a problem-independent random perturbation. For scheduling problems with clear structural characteristics such as DFJSP, the positions of critical paths and bottleneck machines are identifiable in each generation, and changes to these positions by random mutation are purely accidental. How to introduce scheduling structure perception directly into mutation operations, so that gene modification has directional decision-making capability for bottleneck operations, is the core limitation of existing multi-objective evolutionary algorithms in solving MO-DFJSP, and is also the fundamental motivation for designing the CGDRL operator in this paper.

2.3 Application of Graph Neural Networks and Deep Reinforcement Learning in Scheduling

The application of DRL in scheduling problems has evolved from independent end-to-end solvers to auxiliary modules embedded in evolutionary algorithm frameworks. Behind this path lies the continuous deepening of researchers' understanding of the respective advantages and limitations of the two types of technologies.

In the direction of independent solvers, Zhang et al. [29] first combined disjunctive graphs with graph isomorphism networks and proposed an end-to-end DRL framework for job shop scheduling, proving the fundamental value of graph structure representation in capturing scheduling constraint topology. On this basis, Song et al. [30] introduced heterogeneous graph neural networks into FJSP to jointly encode complex relationships between operations and machines, learning high-quality priority dispatching rules in an end-to-end manner. Lei et al. [31] further extended the disjunctive graph into a multi-action DRL framework, using the multi-pointer PPO algorithm to simultaneously decide operation selection and machine assignment, systematically improving the solution quality of DRL on FJSP. However, end-to-end DRL solvers are still inferior to mature evolutionary algorithm frameworks in generalizability and solution quality, and the training cost is high, making it difficult to directly migrate to distributed multi-objective scenarios.

This limitation has prompted researchers to shift toward an ensemble paradigm that embeds reinforcement learning into evolutionary algorithm frameworks. Tao et al. [32] combined NSGA-II with Q-learning for parameter adaptive adjustment in distributed permutation flowshop rescheduling. Zhang et al. [33] proposed a Q-learning hyper-heuristic evolutionary algorithm for DFJSP with crane transportation constraints. Zhang et al. [34] proposed a meta-reinforcement learning-driven metaheuristic algorithm to handle hybrid flowshop scheduling with learning and forgetting effects. Zhang et al. [35] combined Q-learning with multi-population evolutionary algorithm to solve distributed heterogeneous assembly no-idle flowshop scheduling. These works prove that the collaborative integration of reinforcement learning and evolutionary

algorithms can improve convergence behavior, but reinforcement learning always plays the role of a parameter regulator in the above frameworks, while the genetic operations of evolutionary algorithms themselves remain blindly random.

It can be seen that there are two unfilled gaps in existing work. First, the structural perception ability of disjunctive graphs and graph neural networks is only used for state encoding in independent solvers, and has never been introduced into the mutation stage of evolutionary algorithms in the form of gene operators. Second, for the cross-generational population stagnation problem, existing methods lack comprehensive assessment and hierarchical response mechanisms based on multi-dimensional indicators such as hypervolume change rate and objective-space diversity. This paper proposes the CGDRL operator and MSHI strategy to compensate for the above shortcomings from the two aspects of intra-generational gene manipulation and cross-generational state regulation respectively.

3 Problem Description and Model Construction

DFJSP integrates multiple geographically dispersed manufacturing workshops into a unified scheduling framework, extending FJSP to distributed manufacturing environments. This section presents the mathematical modeling of MO-DFJSP, covering the problem description, basic assumptions, and the complete model in sequence.

3.1 Problem Description

The system comprises F geographically dispersed workshops, where workshop f is equipped with M_f machines of differentiated processing capabilities. There are N jobs to be scheduled, and job J_i consists of n_i operations that must be executed in a fixed technological sequence, with the j-th operation of job J_i denoted as O_{ij}. Each operation has processing flexibility: O_{ij} can be assigned to any eligible machine within a candidate workshop set $F_{i,j}$. Consecutive operations belonging to the same job may be processed continuously within the same workshop, or the workpiece may be transferred to another workshop upon completion, incurring corresponding transfer time and energy costs.

The two optimization objectives defined in this paper are inherently conflicting. Minimizing makespan requires compressing the total production cycle and increasing machine utilization, whereas minimizing total energy consumption favors the selection of low-power machines and the reduction of idle standby duration. These two objectives exhibit fundamental tension in machine allocation and operation sequencing decisions, constituting a typical Pareto multi-objective optimization problem.

3.2 Model Construction

To ensure the tractability of the model and accurately reflect real-world production constraints, the following assumptions are made.

(1) All workshops, machines, and jobs are available at time zero.
(2) Each machine processes at most one operation at any given time.

(3) Each operation is assigned to exactly one machine throughout its entire processing duration, and parallel processing across multiple machines is not permitted.
(4) Once an operation commences, it cannot be preempted or interrupted until completion.
(5) The processing time of each operation on each machine, the transfer time between workshops and between machines, and the energy consumption parameters of each machine are all known deterministic quantities.
(6) The loading and unloading time of workpieces is incorporated into the processing time of the corresponding operation and is not modeled separately.
(7) A job cannot simultaneously undergo inter-workshop transfer and intra-workshop machine transfer at the same point in time.

The definitions of all symbols involved in the model are summarized in Table 1.

Table 1 Parameters definitions of the MO-DFJSP mathematical model

Parameters	Description
i, g	Job indices, $i, g \in \{1, \ldots, N\}$
j, h	Operation indices, $j \in \{1, \ldots, n_i\}$
f, r	Workshop indices, $f, r \in \{1, \ldots, F\}$
k, m	Machine indices, $k, m \in \{1, \ldots, M_f\}$
N	Total number of jobs
n_i	Total number of operations of job J_i
F	Total number of workshops
M_f	Total number of machines in workshop f
O_{ij}	The j-th operation of job J_i
F_{ij}	Candidate workshop set eligible to process O_{ij}
M_{ij}^f	Candidate machine set in workshop f eligible to process O_{ij}
S_{ij}	Start time of operation O_{ij}
C_{ij}	Completion time of operation O_{ij}
p_{ij}^{fk}	Processing time of O_{ij} on machine k in workshop f
T_{fr}^c	Transfer time of a workpiece from workshop f to workshop r
T_{mk}^w	Transfer time of a workpiece from machine m to machine k within the same workshop
e_{fk}^p	Unit processing energy consumption of machine k in workshop f
e_{fk}^s	Unit idle energy consumption of machine k in workshop f
e_{tr}	Energy consumption coefficient per unit transfer time
x_{ij}^{fk}	Binary variable, equals 1 if O_{ij} is processed on machine k in workshop f, otherwise 0
$y_{ij,gh}$	Binary variable, equals 1 if O_{ij} immediately precedes O_{gh} on the same machine, otherwise 0
a_i^{mk}	Binary variable, equals 1 if job J_i is transferred from machine m to machine k within the same workshop, otherwise 0
b_i^{fr}	Binary variable, equals 1 if job J_i is transferred from workshop f to workshop r, otherwise 0

A multi-objective optimization model is established with minimizing makespan and minimizing total energy consumption as objectives:

$$\min(Z_1, Z_2) = \min(MT, TE) \tag{1}$$

Objective function Z_1 denotes the makespan, defined as the maximum completion time of the last operation across all jobs:

$$Z_1 = \max\left(C_{i,n_i}\right), \quad \forall i \in \{1, \ldots, N\} \tag{2}$$

Objective function Z_2 denotes the total system energy consumption, comprising three components: processing energy E_p, idle energy E_s, and transfer energy E_t:

$$Z_2 = E_p + E_s + E_t \tag{3}$$

where E_p is the sum of the products of each operation's processing time and the unit processing energy consumption of its assigned machine:

$$E_p = \sum_{f=1}^{F} \sum_{k=1}^{M_f} \sum_{i=1}^{N} \sum_{j=1}^{n_i} p_{ij}^{fk} \cdot e_{fk}^{p} \cdot x_{ij}^{fk} \tag{4}$$

E_s corresponds to the energy consumed by machines in the idle standby state between consecutive operations, accumulated by multiplying the idle interval length between consecutive operations O_{ij} and O_{gh} on the same machine by the unit idle energy consumption:

$$E_s = \sum_{f=1}^{F} \sum_{k=1}^{M_f} \sum_{i=1}^{N} \sum_{j=1}^{n_i} \sum_{g=1}^{N} \sum_{h=1}^{n_g} e_{fk}^{s} \cdot (S_{g,h} - C_{i,j}) \cdot y_{ij,gh} \cdot x_{ij}^{fk} \cdot x_{gh}^{fk} \tag{5}$$

E_t covers both inter-workshop transfer and intra-workshop machine transfer:

$$E_t = \sum_{f=1}^{F} \sum_{r=1}^{F} \sum_{i=1}^{N} e_{tr} \cdot T_{fr}^{c} \cdot b_i^{fr} + \sum_{f=1}^{F} \sum_{m=1}^{M_f} \sum_{k=1}^{M_f} \sum_{i=1}^{N} e_{tr} \cdot T_{mk}^{w} \cdot a_i^{mk} \tag{6}$$

The model is subject to the following constraints:

$$\sum_{f \in F_{i,j}} \sum_{k \in M_{ij}^{f}} x_{ij}^{fk} = 1, \quad \forall i, j \tag{7}$$

$$S_{i,1} \cdot x_{i1}^{fk} \geq 0, \quad \forall i, f, k \tag{8}$$

$$C_{i,j} = \left(S_{i,j} + p_{ij}^{fk}\right) \cdot x_{ij}^{fk} \tag{9}$$

$$S_{i,j} \cdot x_{ij}^{fk} - T_{rf}^{c} \cdot b_i^{rf} - T_{mk}^{w} \cdot a_i^{mk} - C_{i,(j-1)} \cdot x_{i(j-1)}^{rm} \geq 0, \quad \forall i, j, f, r, k, m, \ j \geq 2 \tag{10}$$

$$(S_{g,h} - C_{i,j}) \cdot x_{ij}^{fk} \cdot x_{gh}^{fk} \cdot y_{ij,gh} + (S_{i,j} - C_{g,h}) \cdot x_{ij}^{fk} \cdot x_{gh}^{fk} \cdot y_{gh,ij} \geq 0, \quad \forall i, j, g, h \tag{11}$$

$$a_i^{mk} + b_i^{fr} \leq 1, \quad \forall i, m, k, f, r \tag{12}$$

Constraint (7) stipulates that each operation is processed by exactly one machine, ensuring unique assignment. Constraint (8) ensures that the start time of the first operation of each job is non-negative. Constraint (9) defines the completion time of an operation as the sum of its start time and processing time. Constraint (10) enforces the technological precedence between consecutive operations of the same job, incorporating inter-workshop transfer time T_{rf}^{c} and intra-workshop machine transfer time T_{mk}^{w} into the minimum interval between operations. Constraint (11) guarantees that each machine processes at most one operation at any given time, precluding temporal overlap between any two operations on the same machine. Constraint (12) stipulates that a job cannot simultaneously undergo inter-workshop transfer and intra-workshop machine transfer.

4 Hybrid Algorithm Based on Disjunctive Graph-Enhanced Deep Reinforcement Learning (HENSGA-DRL)

NSGA-II [36] is a classical evolutionary algorithm for multi-objective combinatorial optimization that maintains both convergence and diversity of the Pareto front within a single run through fast non-dominated sorting and crowding distance mechanisms. In the proposed HENSGA-DRL, the global optimization framework adopts NSGA-II as its evolutionary backbone. Existing studies have demonstrated that disjunctive graphs can effectively encode the constraint topology of scheduling solutions, however, this structural perception capability has never been introduced into the mutation stage of evolutionary algorithms in the form of gene operators, and genetic operations themselves remain at the level of random perturbation. To address this gap, this paper designs the Critical-path Graph-guided Deep Reinforcement Learning operator (CGDRL) as a gene operator embedded in the NSGA-II framework. Through a Graph Convolutional Network (GCN), the current scheduling constraint structure of each individual is encoded into operation-level bottleneck features. A policy network trained with Proximal Policy Optimization (PPO) [37] takes these features as state input and generates directed machine reassignment decisions for bottleneck operations on the critical path via online reinforcement learning with a weighted sum of multi-objective normalized improvement rates as the immediate reward, endowing each generation of offspring gene modifications with continuous perception of and adaptive decision-making over the global scheduling structure.

In addition, this paper introduces the Multi-metric State-aware Hierarchical Intervention strategy (MSHI) to address the cross-generational population stagnation problem. MSHI continuously archives historically optimal non-dominated solutions through an Elite Solution Archive (ESA), quantifies population health status in real time using hypervolume change rate and objective-space diversity, and activates three-tier intervention measures according to the stagnation severity gradient. CGDRL and MSHI act on intra-generational gene operation decision-making and cross-generational

population state regulation respectively, collaboratively driving the algorithm to efficiently approximate a high-quality Pareto front.

4.1 Overall Framework

The overall framework of HENSGA-DRL is illustrated in Fig. 1. The global optimization framework adopts NSGA-II as its evolutionary backbone, completing fitness evaluation, offspring generation via crossover and mutation operators, and environmental selection based on non-dominated sorting and normalized crowding distance after merging parent and offspring populations in each iteration, and the iteration terminates when the number of function evaluations reaches the preset limit MaxNFEs. The CGDRL operator is embedded into the mutation stage with probability p_c, sequentially performing disjunctive graph construction, GCN encoder feature extraction, bottleneck operation identification, and PPO policy network machine assignment decisions on selected individuals, with reassignment results written back to chromosomes via MS update, and the remaining $1 - p_c$ proportion of individuals undergo conventional random mutation to maintain global exploration capability.

The MSHI strategy sequentially executes three phases of state management at the end of each generation. The ESA module continuously archives historically optimal non-dominated solutions to provide high-quality solution sources for subsequent intervention. The stagnation detection module quantitatively monitors hypervolume change rate and objective-space diversity, outputting a comprehensive health score Φ_t. The adaptive intervention mechanism takes the stagnation counter SC and Φ_t as triggers, responding to the corresponding intervention tier according to execution priority. Upon algorithm termination, the union of the ESA archive and the first non-dominated layer of the current population is taken as the final Pareto front output.

4.2 Base Framework: NSGA-II

NSGA-II is a classical multi-objective evolutionary algorithm that computes objective function values for all individuals in the population each generation, partitions them into Pareto layers via fast non-dominated sorting, with the first layer constituting the set of mutually non-dominated optimal solutions in the current population. Within the same layer, crowding distance measures the distribution density of individuals in objective space, used to preferentially retain isolated solutions when the number of retained individuals is limited, thereby maintaining diversity of the Pareto front. In the selection phase, the merged parent and offspring population is truncated to size N_p according to non-dominated layers and crowding distance, ensuring that elite individuals are not lost. The specific designs of key components are as follows.

4.2.1 Chromosome Encoding

Each individual is represented by a two-layer vector chromosome of total length $2N_s$, where N_s is the total number of operations. The first N_s positions constitute the Machine Sequence (MS), where each position corresponds to one operation and records the machine index allocated to that operation within its candidate machine set M_{ij}^f, and since each machine uniquely belongs to a specific workshop, the MS layer machine

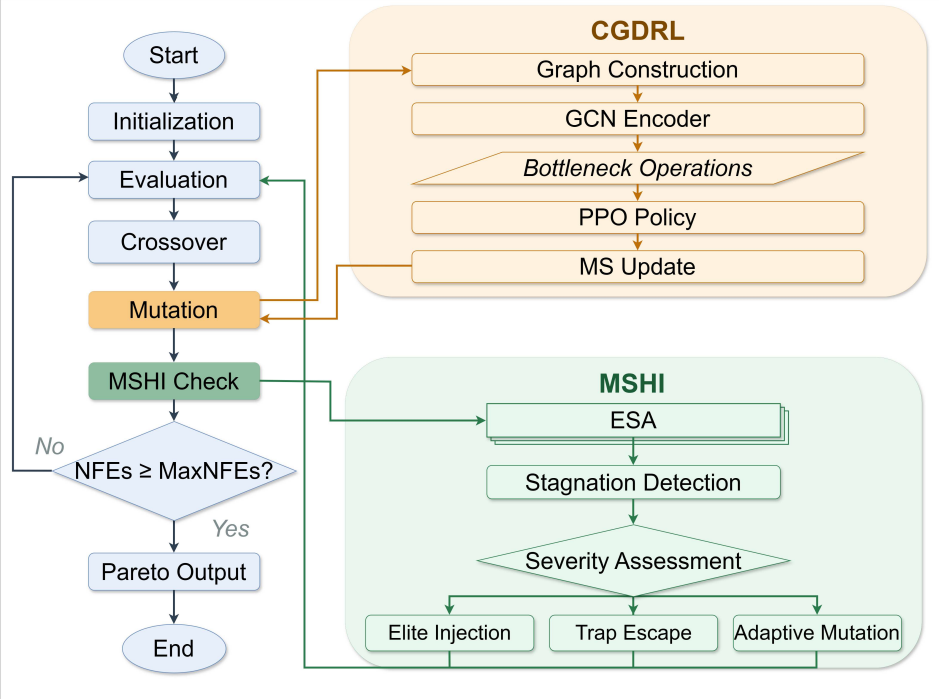


Fig. 1 Overall framework of HENSGA-DRL

assignment decision implicitly determines the processing workshop of each operation, and the machine assignments of consecutive operations of the same job may point to different workshops, with the resulting inter-workshop transfer time incorporated into operation intervals according to constraint (10) during the decoding phase. The subsequent N_s positions constitute the Operation Sequence (OS), forming a permutation of job indices, where the j-th occurrence of job i corresponds to its j-th operation O_{ij}, and this encoding inherently satisfies operation precedence constraints without requiring additional repair.

4.2.2 Hybrid Rule-Based Population Initialization

The quality of the initial population directly influences the convergence speed of the algorithm. This paper adopts a hybrid rule-based initialization strategy, using parameter ρ_r to control the ratio of rule-guided individuals to randomly generated individuals. The rule-guided portion is generated by cycling through nine dispatching rules: SPT, LPT, MFP, LFP, MOP, Fastest, Balanced, LeastLoaded, and Random, where the Random rule uniformly samples from the candidate machine set of each operation, and the randomly generated portion uniformly samples both machine sequences and operation sequences over the complete feasible domain to maintain population diversity at a higher dimension. The complete procedure is shown in Algorithm 1.

Algorithm 1 HybridInit

Require: Population size N_p, problem data, rule ratio ρ_r
Ensure: Initial population P

$n_{rule} \leftarrow \text{round}(N_p \times \rho_r)$; $n_{rand} \leftarrow N_p - n_{rule}$
$P \leftarrow [\,]$
for $i = 0$ **to** $n_{rule} - 1$ **do**
    $rule \leftarrow \text{RULES}[i \bmod |\text{RULES}|]$
    $MS \leftarrow \text{ApplyRule}(rule,\ data)$
    $OS \leftarrow \text{GenerateOS}(data)$
    P.append(concat($MS,\ OS$))
end for
for $i = 0$ **to** $n_{rand} - 1$ **do**
    $MS \leftarrow \text{RandomMS}(data)$
    $OS \leftarrow \text{GenerateOS}(data)$
    P.append(concat($MS,\ OS$))
end for
return P

4.2.3 Genetic Operators

The OS segment adopts Priority Operation Crossover (POX) [38] to preserve the legality of job processing order, while the MS segment adopts uniform crossover. Mutation operates by randomly reassigning machine positions in the MS segment and randomly swapping operation indices in the OS segment, with triggering probability equal to the mutation rate p_m.

4.2.4 Normalized Crowding Distance

Standard NSGA-II computes crowding distance using the raw objective value spacing. Since Z_1 and Z_2 differ significantly in numerical magnitude, direct comparison introduces systematic bias and compromises the distributional uniformity of Pareto front solutions. Accordingly, this paper applies min-max normalization to each objective independently before computing crowding distance. The normalized spacing of individual i in objective q direction is defined as:

$$d_q(i) = \frac{f_q(i+1) - f_q(i-1)}{f_q^{\max} - f_q^{\min}} \tag{13}$$

where $f_q(i+1)$ and $f_q(i-1)$ are the objective values of the right and left neighbors of individual i sorted by objective q within the same non-dominated layer, and $f_q^{\max}$ and $f_q^{\min}$ are the boundary values of that layer in objective q direction. The comprehensive normalized crowding distance of individual i is:

$$\text{NCD}(i) = \sum_{q=1}^{2} d_q(i) \tag{14}$$

The NCD of boundary individuals is set to positive infinity to guarantee their preferential retention, thereby preserving the completeness of Pareto front endpoints. This

normalization eliminates the dimensional discrepancy between Z_1 and Z_2, enabling crowding distance to fairly measure the true distribution density of solutions in multi-objective space.

4.3 Critical-Path Graph-Guided Deep Reinforcement Learning Operator (CGDRL)

The CGDRL operator is the core enhancement component of HENSGA-DRL, translating the random search over machine assignment genes into structure-guided directed optimization by parsing chromosomes into disjunctive graphs and utilizing graph neural networks to identify bottleneck operations. The operator acts on the current population with probability p_c at each evolutionary generation, performing graph-guided directed machine reassignment on selected individuals and conventional random mutation on the remaining $1 - p_c$ proportion. The complete execution procedure is shown in Algorithm 2.

Algorithm 2 CGDRL Operator

Require: Chromosome (MS, OS), mutation rate p_m, selection ratio α, policy parameters θ, replay buffer $\mathcal{D}$
Ensure: Mutated chromosome MS', updated policy parameters θ

$G \leftarrow \text{BuildDisjunctiveGraph}(MS,\ OS)$
$X \leftarrow \text{ExtractNodeFeatures}(G)$; $A \leftarrow \text{GetAdjacencyMatrix}(G)$
$\mathbf{h}_g,\ H_{nodes} \leftarrow \text{GCNEncoder}(X,\ A)$
$scores \leftarrow \text{BottleneckDetector}(H_{nodes},\ \mathbf{h}_g)$ (Eq. (15))
$k \leftarrow \max(1,\ \lfloor N_s \times \alpha \times p_m \rfloor)$; $S_b \leftarrow \text{TopK}(scores,\ k)$
$\mathbf{L} \leftarrow \text{NormalizedMachineLoad}(MS)$; $MS' \leftarrow \text{copy}(MS)$; $buffer \leftarrow [\,]$
for each operation op in S_b **do**
    $\mathbf{s} \leftarrow [H_{nodes}[op];\ \mathbf{L}]$ (Eq. (16))
    $machine \leftarrow \text{ActorNetwork}(\mathbf{s},\ \theta)$
    $r \leftarrow \text{RewardFunction}(MS',\ op,\ machine)$ (Eqs. (17)–(18))
    $MS'[op] \leftarrow machine$; $\text{UpdateLoad}(\mathbf{L},\ op,\ machine)$
    $buffer.\text{add}((\mathbf{s},\ machine,\ r))$
end for
$\mathcal{D}.\text{add}(buffer)$
if $|\mathcal{D}| \geq batch_size$ **then**
    $\theta \leftarrow \text{PPO_Update}(\mathcal{D},\ \theta)$
end if
return MS', θ

4.3.1 Disjunctive Graph Construction

Encoding a chromosome as a disjunctive graph $G = (V,\ E_c \cup E_d)$ is the foundation for CGDRL to extract scheduling structural information. E_c encodes the intra-job technological precedence constraints as static directed edges, with edge weights carrying

the processing time of predecessor operations, enabling direct quantification of critical path length on the graph. E_d encodes machine resource contention as dynamic directed edges, rebuilt as MS assignments change, allowing the graph representation to perceive the quality of the current solution. For each node, a six-dimensional feature vector $\mathbf{x}_v$ is constructed from six normalized attributes: processing time, energy consumption, relative position within the job, proportion of eligible machines, normalized load of the assigned machine, and whether the operation lies on the critical path. Fig. 2 presents a scheduling instance with four jobs and four machines, where O_{22} incurs a waiting delay due to resource contention with O_{12} over M_2, resulting in $Z_1 = 24$, and this example serves as the running illustration throughout Section 4.3.

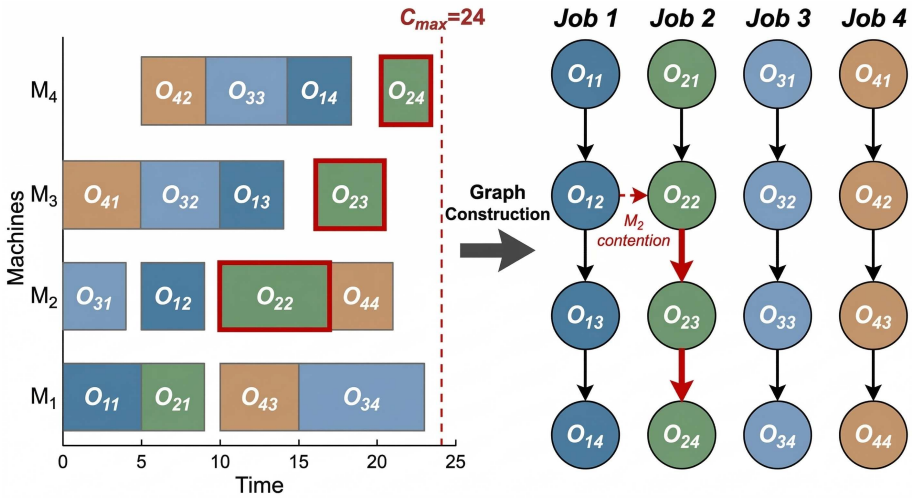


Fig. 2 Example of disjunctive graph construction

4.3.2 GCN Encoding and Bottleneck Detection

The node feature vectors $\mathbf{x}_v$ are stacked to form the feature matrix $X \in \mathbb{R}^{|V|\times 6}$, and after three-layer GCN aggregation of neighbor information, the node embedding matrix H_{nodes} is obtained, with three-layer propagation enabling each operation's embedding to simultaneously capture local machine load and distant critical path layout. The graph-level embedding $\mathbf{h}_g$ is obtained by mean pooling as a global summary of the current scheduling solution. The bottleneck detector uses $\mathbf{h}_g$ as query and H_{nodes} as key-value, quantifying the association strength between each operation and the global scheduling state via a multi-head attention mechanism, and after adding the enhanced representation and original embedding as a residual, a sigmoid-activated MLP compresses the result to a scalar bottleneck score:

$$s_v = \sigma(\mathrm{MLP}(\mathbf{h}_v + \mathrm{Attn}(\mathbf{h}_g,\ H_{nodes},\ H_{nodes}))), \quad s_v \in [0,1] \tag{15}$$

The top-k operations with the highest bottleneck scores form the bottleneck candidate set S_b, where $k = \max(1, \lfloor N_s \times \alpha \times p_m \rfloor)$, with α as the selection ratio and

p_m as the current generation mutation rate, enabling the candidate set size to adaptively shrink with the evolutionary stage and concentrate on operations with the greatest improvement potential. As shown in Fig. 3, the selection threshold is dynamically determined by the k-th highest bottleneck score, and O_{22} ranks first among all operations and uniquely enters S_b.

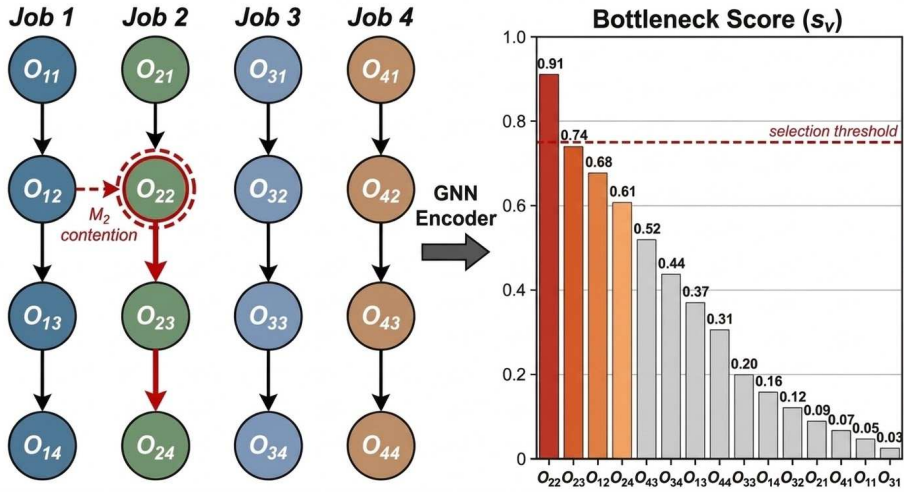


Fig. 3 Example of bottleneck detection results

4.3.3 Action Policy and Reward Function

After the bottleneck candidate set S_b is determined, the Actor network recommends the optimal machine reassignment target for each candidate operation. As illustrated in Fig. 4 using operation O_{22} as an example, the state vector takes the node embedding $\mathbf{h}_{O_{ij}} \in \mathbb{R}^{d_h}$ and the normalized machine load vector $\mathbf{L} \in \mathbb{R}^{M_{total}}$ as inputs, and the complete state vector is defined as:

$$\mathbf{s} = \left[\mathbf{h}_{O_{ij}};\ \mathbf{L}\right] \in \mathbb{R}^{d_h + M_{total}} \tag{16}$$

where d_h is the GCN node embedding dimension and M_{total} is the total number of machines globally. This design enables the decision to simultaneously perceive the structural position of the operation and the current global load distribution. The network outputs candidate machine selection probabilities via a fully connected layer with temperature-scaled softmax, where temperature τ decays linearly with generation count to balance exploration and exploitation, and $-\infty$ masking is applied to ineligible machines. In the example of Fig. 4, M_2 is masked due to the current assignment, while M_4 obtains the highest sampling probability due to the lightest load and is determined as the recommended target for O_{22}.

Policy training uses PPO to constrain the parameter update magnitude, combined with Generalized Advantage Estimation (GAE) to reduce policy gradient variance.

The immediate reward is defined as the weighted sum of multi-objective normalized improvement rates, and the relative improvement rate of objective q is:

$$\rho_q = \frac{Z_q^0 - Z_{q'}}{Z_q^0}, \quad q \in \{1, 2\} \tag{17}$$

where Z_q^0 and $Z_{q'}$ denote the q-th objective value before and after reassignment respectively. The immediate reward is:

$$r = w_1\rho_1 + w_2\rho_2 \tag{18}$$

with weight coefficients satisfying $w_1 + w_2 = 1$. A positive reward when $r > 0$ and a negative reward otherwise guides the policy to internalize the trade-off relationship rather than biasing toward a single objective. As shown at the bottom of Fig. 4, the reward signal r generated after reassignment is fed back to drive policy parameter updates, continuously reinforcing the decision tendency to migrate critical path operations to low-load machines.

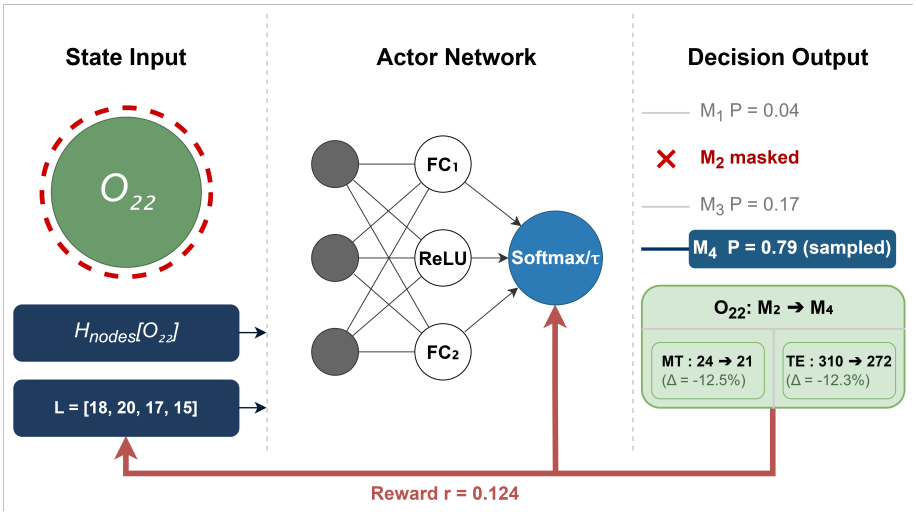


Fig. 4 Example of Actor network decision-making

The Gantt chart comparison before and after rescheduling is shown in Fig. 5, where after migrating O_{22} from M_2 to M_4, the machine load distribution becomes more balanced and Z_1 is compressed from 24 to 21, validating the improvement effect of the CGDRL operator's structure-aware directed reassignment on makespan.

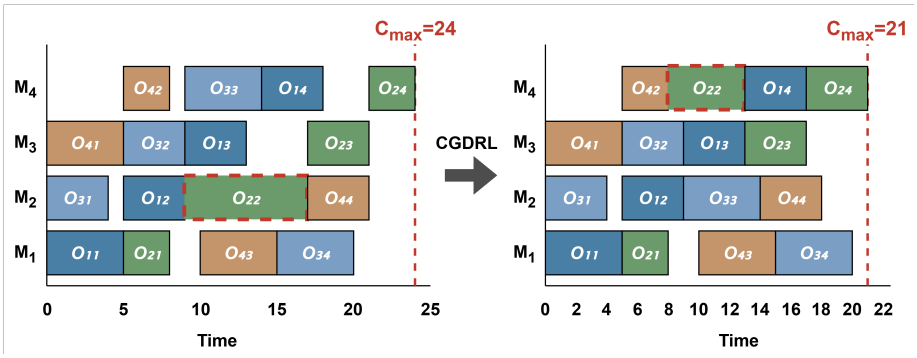


Fig. 5 Gantt chart comparison before and after rescheduling

4.4 Multi-Metric State-Aware Hierarchical Intervention Strategy (MSHI)

The Elite Solution Archive continuously preserves historically optimal non-dominated solutions to provide high-quality solution sources for intervention operations. The Multi-metric Performance Tracker (MPT) quantifies the population health state at the end of each generation and outputs a comprehensive health score Φ_t. The Adaptive Intervention Mechanism (AIM) takes Φ_t and stagnation counter SC as triggers, responding to the corresponding intervention tier according to execution priority. The three sub-modules are invoked sequentially at the end of each generation, jointly driving the algorithm to continuously approximate the global Pareto front.

4.4.1 Elite Solution Archive (ESA)

ESA is maintained independently of the main population and preserves the historically optimal non-dominated solution set discovered throughout the entire evolutionary process, with a capacity limit of A individuals. At the end of each generation, the current population and existing archive solutions are merged, non-dominated sorting is performed, the first-layer front is retained, and when capacity is exceeded, truncation is performed in descending order of crowding distance:

$$\mathrm{ESA}_t = \mathrm{Trunc}(\mathrm{NDS}(\mathrm{ESA}_{t-1} \cup P_t),\ A) \tag{19}$$

where $\mathrm{NDS}(\cdot)$ denotes the non-dominated filtering operation and $\mathrm{Trunc}(\cdot,\ A)$ retains the A most sparsely distributed solutions based on crowding distance. ESA exposes two interfaces to elite injection, returning respectively the individual with the best comprehensive objective value and the individual with the largest crowding distance, and the two types of solutions are mixed equally during injection to maintain injection direction diversity while introducing high-quality solutions.

4.4.2 Multi-Metric Performance Tracker (MPT)

MPT performs weighted fusion of three metrics at the end of each generation to compute the comprehensive health score:

$$\Phi_t = \lambda_1 \cdot \frac{D_t}{D_{\max}} + \lambda_2 \cdot \frac{HV_t}{HV_{\max}} - \lambda_3 \cdot \frac{SC}{T} \tag{20}$$

where D_t is the average nearest-neighbor distance of the current non-dominated solutions in objective space, measuring the coverage uniformity of the Pareto front. HV_t is the objective-space volume covered by the Pareto front relative to a reference point, comprehensively reflecting convergence and diversity. SC is the number of consecutive non-improving generations, incremented when the inter-generational hypervolume increment $\Delta HV < \delta_{hv}$ and reset to zero otherwise. T is the maximum number of evolutionary generations, normalizing the stagnation penalty term to $[0, 1]$. λ_1, λ_2, λ_3 are weight coefficients, with the convergence quality term carrying the highest weight and the stagnation penalty term applying negative adjustment. Φ_t serves as the unified triggering signal for AIM's three-tier intervention decisions.

4.4.3 Adaptive Intervention Mechanism (AIM)

AIM divides intervention intensity into three tiers, deeper stagnation activates higher priority, and when a high-priority tier is activated, lower-priority tiers suspend response. The complete execution procedure is shown in Algorithm 3.

The first tier is dynamic mutation rate scheduling, which operates as background intervention throughout the entire evolutionary cycle without requiring stagnation triggering. The adjustment rule for mutation rate $p_m(t)$ with the evolutionary process is:

$$p_m(t) = \begin{cases} p_m^0, & t \le t_d \\ p_m^0 - \left(p_m^0 - p_m^1\right) \cdot \dfrac{t - t_d}{T - t_d}, & t > t_d \end{cases} \tag{21}$$

where p_m^0 is the initial mutation rate, p_m^1 is the terminal mutation rate, t_d is the decay start generation, and T is the maximum number of evolutionary generations. For the first t_d generations, the mutation rate is maintained at p_m^0 to reinforce global exploration, and thereafter it decays linearly with generation count, gradually transitioning the population from global exploration to fine-grained local exploitation in the convergence phase.

The second tier is elite injection, triggered when accumulated stagnation reaches the injection threshold $SC \ge \delta$ and the health score falls below the alert level $\Phi_t < \theta_h$. Historically optimal solutions are extracted from ESA to replace an equal number of individuals with the worst objective values in the population, and a cooling period T_{cd} is set after injection to prevent consecutive injections from suppressing the normal evolutionary rhythm.

The third tier is trap escape, forcibly triggered at the highest priority when stagnation further accumulates to $SC \ge \tau_s$ and the current generation exceeds the detection start generation t_s, directly elevating the mutation rate to the escape level p_m^e, overriding the background scheduling of Eq. (21) for t_e generations before recovery, with SC reset to zero for fresh accumulation.

Algorithm 3 MSHI Strategy

Require: Population P, objective values Obj, elite archive ESA, stagnation counter SC, current generation t, escape flag ξ, escape counter c_e
Ensure: Updated P, SC, ξ

$\text{ESA}_t \leftarrow \text{Trunc}(\text{NDS}(\text{ESA}_{t-1} \cup P),\ A)$ (Eq. (19))
$D_t,\ HV_t \leftarrow \text{ComputeMetrics}(P,\ Obj)$
$SC \leftarrow SC + 1$ **if** $HV_t - HV_{t-1} < \delta_{hv}$ **else** 0
$\Phi_t \leftarrow$ Eq. (20)
// Tier 3 (highest priority): trap escape
if $SC \geq \tau_s$ **and** $t > t_s$ **then**
    $\xi \leftarrow \text{True}$; $c_e \leftarrow 0$
end if
if $\xi = \text{True}$ **then**
    $p_m^* \leftarrow p_m^e$
    $c_e \leftarrow c_e + 1$
    if $c_e \geq t_e$ **then**
        $\xi \leftarrow \text{False}$; $SC \leftarrow 0$
    end if
    return $P,\ SC,\ \xi$
end if
// Tier 2: elite injection
if $SC \geq \delta$ **and** $\Phi_t < \theta_h$ **and** $cooldown = 0$ **then**
    $elites \leftarrow \text{ESA.GetBest}(k) \cup \text{ESA.GetDiverse}(k)$
    $P \leftarrow \text{ReplaceWorst}(P,\ Obj,\ elites)$; $cooldown \leftarrow T_{cd}$
end if
// Tier 1 (background): dynamic mutation rate
$p_m^* \leftarrow$ Eq. (21); $cooldown \leftarrow \max(0,\ cooldown - 1)$
return $P,\ SC,\ \xi$

5 Experiments and Analysis

This section presents a systematic validation of HENSGA-DRL through ablation and comparative experiments. All experiments were conducted on the same hardware platform configured with an AMD Ryzen 5 5600X CPU @ 3.7 GHz, 32 GB RAM, and an NVIDIA GeForce RTX 3050 GPU, with Python 3.10 as the software environment, ensuring fairness of comparison across all algorithms.

5.1 Benchmark Dataset

Experiments are conducted on the DLA benchmark dataset, constructed by extending the standard LA benchmark instances, comprising 20 test instances (DLA01–DLA20). The original LA series targets single-workshop environments without energy consumption modeling, and this paper extends it by introducing distributed workshop structures, machine working power, idle power, and inter-workshop transportation time parameters. The detailed parameters of each instance are listed in Table 2.

Table 2 Detailed information of the DLA dataset

Instance	Workshops	Jobs	Operations	Machines	Source
DLA01	2	10	47	5	LA01
DLA02	3	10	47	5	LA02
DLA03	3	10	47	5	LA03
DLA04	4	10	47	5	LA04
DLA05	4	10	47	5	LA05
DLA06	2	15	71	5	LA06
DLA07	3	15	71	5	LA07
DLA08	3	15	71	5	LA08
DLA09	4	15	71	5	LA09
DLA10	4	15	71	5	LA10
DLA11	2	20	95	5	LA11
DLA12	3	20	95	5	LA12
DLA13	3	20	95	5	LA13
DLA14	4	20	95	5	LA14
DLA15	4	20	95	5	LA15
DLA16	2	10	95	10	LA16
DLA17	3	10	95	10	LA17
DLA18	3	10	95	10	LA18
DLA19	4	10	95	10	LA19
DLA20	4	10	95	10	LA20

5.2 Performance Metrics

Solution quality is directly assessed by the best mean value of each optimization objective on the Pareto front, where the two objectives are makespan (MT) and total energy consumption (TE), with lower values indicating higher solution quality. To comprehensively measure algorithm performance in both convergence and diversity dimensions, particularly when the true Pareto front is unknown, this paper adopts hypervolume (HV) as the primary comprehensive performance indicator, computed as:

$$\mathrm{HV}(P, \mathbf{r}) = \lambda\left(\bigcup_{\{\mathbf{x}\in P\}} v(\mathbf{x}, \mathbf{r})\right) \tag{22}$$

where P is the non-dominated solution set obtained by the algorithm, $\mathbf{r}$ is the preset reference point, $v(\mathbf{x}, \mathbf{r})$ is the hyperrectangle volume enclosed by solution $\mathbf{x}$ and reference point $\mathbf{r}$, and $\lambda(\cdot)$ denotes the Lebesgue measure. A larger HV value indicates superior convergence and diversity of the algorithm.

5.3 Parameter Configuration

The parameters of HENSGA-DRL are determined through preliminary experiments to achieve a reasonable balance between computational efficiency and optimization performance. The final parameter configuration is listed in Table 3. The parameters of MOEA/D, SPEA-II, and HMOMA are set according to the recommended values in their respective original papers. MaxNFEs is uniformly set to 80000 (corresponding to $T = 400$ evolutionary generations) for all algorithms to ensure fairness of comparison.

Table 3 Main parameter configuration of HENSGA-DRL

Parameter	Value
Population size N_p	200
Crossover probability P_c	0.9
Initial mutation rate p_m^0	0.35
Terminal mutation rate p_m^1	0.15
MaxNFEs	80000
Maximum generations T	400
Trap escape threshold τ_s	30
CGDRL trigger probability p_c	0.15
Elite archive capacity A	200
PPO clipping coefficient ε	0.2
Discount factor γ	0.95
Learning rate η	0.001

5.4 Ablation Study

The primary enhancement components of HENSGA-DRL are the CGDRL operator and the MSHI strategy. To validate the individual contribution of each, three ablation variants are designed,Variant A is the pure NSGA-II baseline with all enhancement modules removed, Variant B integrates only the MSHI strategy, and Variant C integrates only the CGDRL operator. All variants adopt the same parameter configuration as HENSGA-DRL and are executed 20 independent runs per instance.

Analysis of Table 4 reveals that Variant B degrades by an average of 10.3% and 9.2% relative to HENSGA-DRL on MT and TE respectively, while the corresponding degradation of Variant C is 6.4% and 5.5%,the former exceeds the latter at the overall mean level, indicating that critical-path graph-guided directed reassignment is the primary contributor to convergence accuracy. The degradation of Variant C is also non-negligible, demonstrating that elite preservation and hierarchical intervention play an important supporting role in maintaining the exploration capability of the CGDRL operator.

For further analysis of multi-objective space coverage, Table 5 shows that HENSGA-DRL achieves the highest HV value on all 20 instances, and on the vast majority of instances the HV improvement magnitude follows the module introduction order. As clearly visible in Fig. 6, the overall distribution of the HENSGA-DRL box is markedly shifted upward relative to other algorithms, its median exceeds that of all variants, and its box is narrower, demonstrating that the performance advantage is stable across runs. All 60 pairwise Wilcoxon rank-sum test comparisons yield $p < 0.05$, confirming the statistical significance of each module's contribution.

Table 4 Comparison results of the ablation study on MT and TE metrics

		HENSGA-DRL		A		B		C	
Instance	Size	MT	TE	MT	TE	MT	TE	MT	TE
DLA01	10×5	609.0	72337.3	**601.0**	**71957.8**	670.6	78267.0	672.8	79186.2
DLA02	10×5	567.1	61731.6	**558.8**	**61381.6**	619.3	66927.0	590.0	64781.0
DLA03	10×5	**564.5**	**61706.4**	672.7	69897.3	611.7	66252.2	590.5	64205.2
DLA04	10×5	**595.5**	**63230.1**	697.7	71922.6	650.4	67832.6	622.8	65590.3
DLA05	10×5	**503.0**	**57321.3**	589.9	66900.3	545.2	61923.3	529.1	59094.2
DLA06	15×5	**833.0**	**101820.3**	980.9	117116.9	909.1	111898.2	940.4	113453.5
DLA07	15×5	**778.0**	**93966.2**	922.1	107672.1	858.7	102310.7	811.0	97715.8
DLA08	15×5	**845.0**	**96320.1**	990.1	115060.1	925.4	105302.0	884.1	100211.6
DLA09	15×5	**888.0**	**107268.2**	1063.2	123475.5	985.1	117512.6	940.2	112406.0
DLA10	15×5	**763.4**	**102534.3**	916.0	122560.5	833.5	111138.9	794.6	106277.5
DLA11	20×5	**1106.0**	**136793.0**	1330.9	159910.3	1234.9	150080.4	1225.9	151331.5
DLA12	20×5	**960.0**	**117757.4**	1140.9	142033.5	1057.4	130679.8	1013.2	121758.1
DLA13	20×5	**1053.0**	**131283.4**	1242.9	158469.9	1173.2	145013.0	1111.4	135416.3
DLA14	20×5	**1123.0**	**138087.8**	1349.3	159921.0	1254.7	152237.1	1175.4	142406.0
DLA15	20×5	**1147.0**	**138537.0**	1374.9	162471.0	1278.4	152771.8	1292.2	152967.9
DLA16	10×10	**940.5**	**138016.2**	1116.5	165609.4	1048.9	151987.0	992.6	143519.7
DLA17	10×10	**744.5**	**118966.0**	901.9	140371.3	819.3	129929.6	774.7	124049.2
DLA18	10×10	**877.0**	**133708.6**	1051.4	158524.9	980.6	146409.0	919.3	139740.3
DLA19	10×10	**847.5**	**135000.4**	1011.7	155977.9	937.2	147585.7	897.2	141232.3
DLA20	10×10	**883.5**	**139298.1**	1076.1	168690.3	986.1	152354.7	995.5	154088.8

Bold values indicate the best result on each instance.

Table 5 HV comparison and Wilcoxon test results of the ablation study

		HENSGA-DRL	A		B		C	
Instance	Size	HV	HV	p-val	HV	p-val	HV	p-val
DLA01	10×5	**0.6955**	0.6907	0.04625	0.6507	0.03319	0.6464	0.02619
DLA02	10×5	**0.7606**	0.7540	0.04396	0.7048	0.01091	0.7283	0.03766
DLA03	10×5	**0.7557**	0.6697	0.00371	0.7111	0.03386	0.7302	0.04003
DLA04	10×5	**0.7311**	0.6538	0.00479	0.6861	0.01790	0.7074	0.03606
DLA05	10×5	**0.8164**	0.7200	0.00337	0.7663	0.01202	0.7900	0.04066
DLA06	15×5	**0.6414**	0.5365	0.00021	0.5779	0.00937	0.5632	0.00262
DLA07	15×5	**0.6996**	0.5855	0.00050	0.6284	0.00184	0.6675	0.03729
DLA08	15×5	**0.6561**	0.5437	0.00050	0.5909	0.00368	0.6241	0.03830
DLA09	15×5	**0.5998**	0.5097	0.00031	0.5399	0.00573	0.5660	0.03371
DLA10	15×5	**0.6689**	0.5446	0.00006	0.6063	0.00691	0.6399	0.03938
DLA11	20×5	**0.5055**	0.4133	0.00010	0.4454	0.00294	0.4450	0.00786
DLA12	20×5	**0.6101**	0.4835	0.00013	0.5362	0.00020	0.5772	0.02696
DLA13	20×5	**0.5359**	0.4293	0.00005	0.4690	0.00010	0.5073	0.03283
DLA14	20×5	**0.4975**	0.4105	0.00053	0.4377	0.00076	0.4740	0.04278
DLA15	20×5	**0.4899**	0.4041	0.00053	0.4322	0.00095	0.4293	0.00262
DLA16	10×10	**0.5015**	0.4212	0.00035	0.4491	0.00192	0.4760	0.03478
DLA17	10×10	**0.6235**	0.5081	0.00017	0.5613	0.00431	0.5952	0.04058
DLA18	10×10	**0.5315**	0.4397	0.00047	0.4742	0.00300	0.5043	0.02983
DLA19	10×10	**0.5380**	0.4510	0.00049	0.4829	0.00255	0.5075	0.02761
DLA20	10×10	**0.5160**	0.4231	0.00007	0.4624	0.00234	0.4574	0.01400

Bold values indicate the best HV on each instance, smaller p-values indicate more significant differences ($p < 0.05$ is significant).

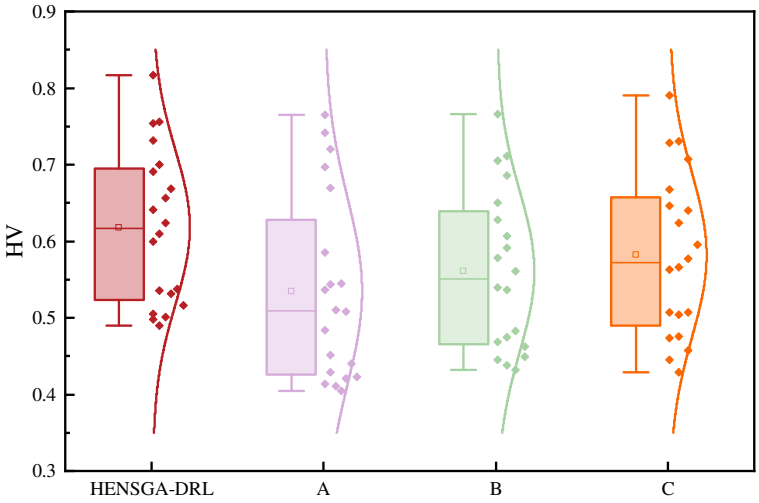


Fig. 6 HV distribution box plots of the four algorithms in the ablation study

5.5 Comparative Study

To validate the competitiveness of HENSGA-DRL, it is compared against three state-of-the-art algorithms: MOEA/D [39], SPEA-II [40], and HMOMA [41]. All three belong to advanced frameworks for multi-objective scheduling, share a similar evolutionary structure with the proposed method, and HMOMA addresses a distributed multi-objective scheduling scenario that partially overlaps with the present study. Each algorithm is executed 20 independent runs per instance with a unified MaxNFEs budget. MT and TE results are reported in Table 6, and HV together with Wilcoxon rank-sum test results are reported in Table 7.

Analysis of Table 6 shows that HENSGA-DRL outperforms all three baseline algorithms on MT and TE across the vast majority of instances. The gap against HMOMA is most pronounced, with average improvements of 14.8% and 13.1% on MT and TE respectively, the improvement margins against MOEA/D are approximately 8.5% on MT and 9.5% on TE, and those against SPEA-II are approximately 9.0% and 9.6%, with both pairs yielding similar margins in the range of 8%–10%. The improvement magnitude generally increases with problem scale, which is especially pronounced on the high-flexibility instances DLA16–DLA20. The Pareto fronts of the four algorithms on representative instances are presented in Fig. 7, where the solution quality achieved by HENSGA-DRL is clearly superior to that of the other algorithms.

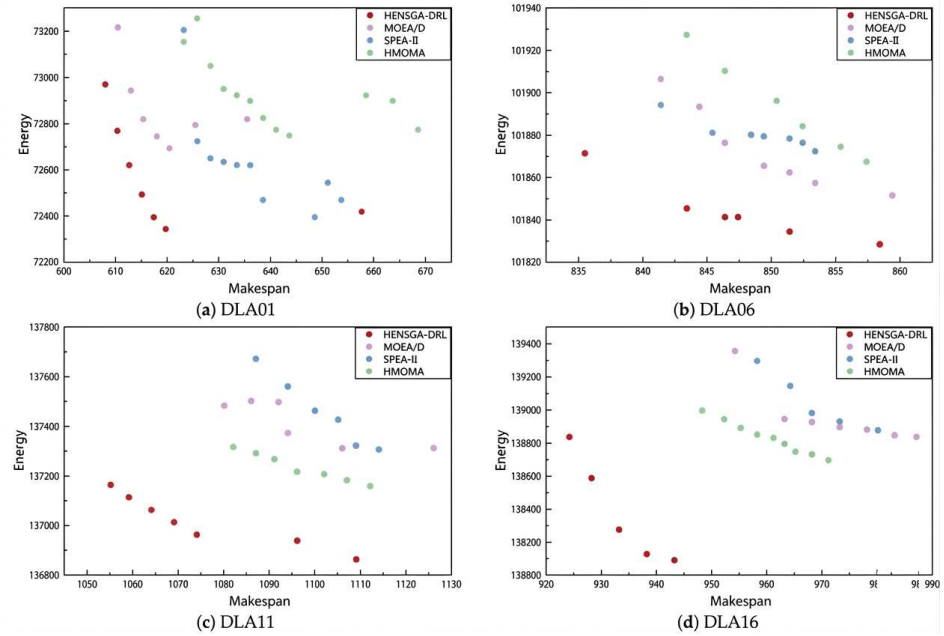


Fig. 7 Pareto front distribution comparison on representative instances

Table 7 provides the HV-based results of the four algorithms, with the corresponding box plots shown in Fig. 8. HENSGA-DRL achieves the highest HV value on all

Table 6 Comparison results of the comparative study on MT and TE metrics

		HENSGA-DRL		MOEA/D		SPEA-II		HMOMA	
Instance	Size	MT	TE	MT	TE	MT	TE	MT	TE
DLA01	10×5	**610.8**	**72373.5**	669.5	78736.7	665.5	77678.6	688.3	80847.3
DLA02	10×5	**568.8**	**61762.5**	558.8	67149.1	614.4	67032.2	644.2	68487.5
DLA03	10×5	**566.2**	**61737.3**	615.3	67223.2	617.0	67019.8	646.1	68809.0
DLA04	10×5	**597.3**	**63261.7**	656.7	68907.7	649.6	68633.7	674.4	70308.6
DLA05	10×5	**504.5**	**57349.9**	545.2	62063.6	496.8	61453.0	569.3	65168.4
DLA06	15×5	**835.5**	**101871.2**	913.1	111018.1	935.4	110760.6	961.0	116180.5
DLA07	15×5	**780.3**	**94013.2**	854.2	102903.5	859.1	103317.6	896.6	106369.4
DLA08	15×5	**847.5**	**96368.3**	925.3	105687.7	931.9	104617.0	958.7	109246.5
DLA09	15×5	**890.7**	**107321.8**	982.9	115943.1	984.5	116636.8	1023.7	119875.7
DLA10	15×5	**765.7**	**102585.6**	845.2	111585.9	856.1	111215.7	873.1	114335.6
DLA11	20×5	**1109.3**	**136861.4**	1118.9	151580.4	1228.8	151888.7	1300.9	156324.2
DLA12	20×5	**962.9**	**117816.3**	1063.8	128639.1	950.4	131602.3	1122.1	133552.0
DLA13	20×5	**1056.2**	**131349.0**	1177.2	145527.1	1189.9	146243.5	1231.2	147552.9
DLA14	20×5	**1126.4**	**138156.8**	1242.7	153073.7	1259.5	150629.0	1288.7	158402.3
DLA15	20×5	**1150.4**	**138606.3**	1265.6	151526.6	1284.4	153957.9	1341.5	156482.5
DLA16	10×10	**943.3**	**138085.2**	1051.1	151168.1	1046.8	153605.3	1099.9	159346.0
DLA17	10×10	**746.7**	**119025.5**	831.2	131090.7	746.3	131050.8	859.2	134470.2
DLA18	10×10	**879.6**	**133775.5**	974.5	148202.0	984.2	149005.1	1020.4	152915.6
DLA19	10×10	**850.0**	**135067.9**	836.0	147751.4	953.4	148361.7	969.8	154763.4
DLA20	10×10	**886.2**	**139367.7**	991.5	154569.7	999.1	153458.4	1010.8	161266.9

Bold values indicate the best result on each instance.

20 instances, with its distribution densely concentrated in the high-HV region, its median exceeds that of all comparison algorithms, and the HMOMA box is the widest, exhibiting the largest performance variability. All 60 Wilcoxon rank-sum test p-values are below 0.05, establishing that HENSGA-DRL is statistically significantly superior to all three state-of-the-art algorithms.

Table 7 HV comparison and Wilcoxon test results of the comparative study

		HENSGA-DRL	MOEA/D		SPEA-II		HMOMA	
Instance	Size	HV	HV	p-val	HV	p-val	HV	p-val
DLA01	10×5	**0.6798**	0.6383	0.01415	0.6423	0.01969	0.6281	0.01945
DLA02	10×5	**0.7460**	0.7271	0.02586	0.6975	0.00521	0.6769	0.00635
DLA03	10×5	**0.7478**	0.6963	0.01082	0.6961	0.01455	0.6750	0.00645
DLA04	10×5	**0.7216**	0.6694	0.00698	0.6737	0.01449	0.6571	0.01000
DLA05	10×5	**0.8133**	0.7598	0.00949	0.7945	0.02736	0.7304	0.00798
DLA06	15×5	**0.6316**	0.5672	0.00326	0.5599	0.00336	0.5374	0.00114
DLA07	15×5	**0.6947**	0.6189	0.00458	0.6151	0.00278	0.5882	0.00031
DLA08	15×5	**0.6473**	0.5783	0.00364	0.5785	0.00221	0.5553	0.00053
DLA09	15×5	**0.5875**	0.5315	0.00435	0.5296	0.00228	0.5129	0.00106
DLA10	15×5	**0.6610**	0.5891	0.00173	0.5867	0.00433	0.5696	0.00078
DLA11	20×5	**0.4940**	0.4573	0.01470	0.4314	0.00076	0.4116	0.00032
DLA12	20×5	**0.6062**	0.5315	0.00060	0.5604	0.01209	0.4986	0.00017
DLA13	20×5	**0.5261**	0.4552	0.00072	0.4506	0.00070	0.4388	0.00043
DLA14	20×5	**0.4857**	0.4267	0.00049	0.4275	0.00152	0.4107	0.00008
DLA15	20×5	**0.4777**	0.4249	0.00447	0.4177	0.00041	0.4049	0.00045
DLA16	10×10	**0.4901**	0.4388	0.00399	0.4365	0.00183	0.4204	0.00123
DLA17	10×10	**0.6201**	0.5462	0.00336	0.5777	0.01371	0.5266	0.00041
DLA18	10×10	**0.5214**	0.4613	0.00482	0.4575	0.00034	0.4430	0.00010
DLA19	10×10	**0.5281**	0.4963	0.01127	0.4661	0.00250	0.4500	0.00148
DLA20	10×10	**0.5050**	0.4461	0.00375	0.4465	0.00290	0.4317	0.00150

Bold values indicate the best HV on each instance, smaller p-values indicate more significant differences ($p < 0.05$ is significant).

The Gantt charts in Fig. 9 ground the above conclusions at the level of concrete scheduling solutions. The scheduling results on DLA01, DLA06, and DLA11 all exhibit balanced machine loads and compact operation sequencing. For DLA16, the doubled number of machines allows operations to be flexibly distributed over a larger resource pool, reducing the average load per machine, and it is precisely this dispersed allocation that enables the multi-objective trade-off space to be more fully exploited, and the algorithm's objective function performance on this instance remains consistent with instances of other scales.

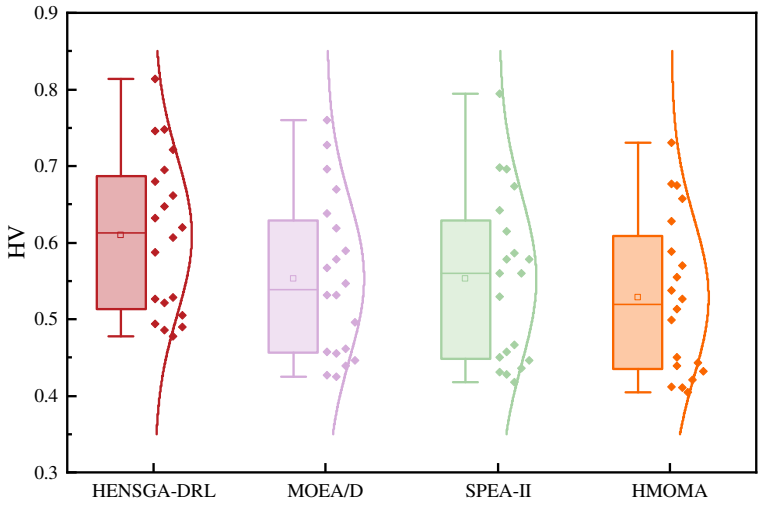


Fig. 8 HV distribution box plots of the four algorithms in the comparative study

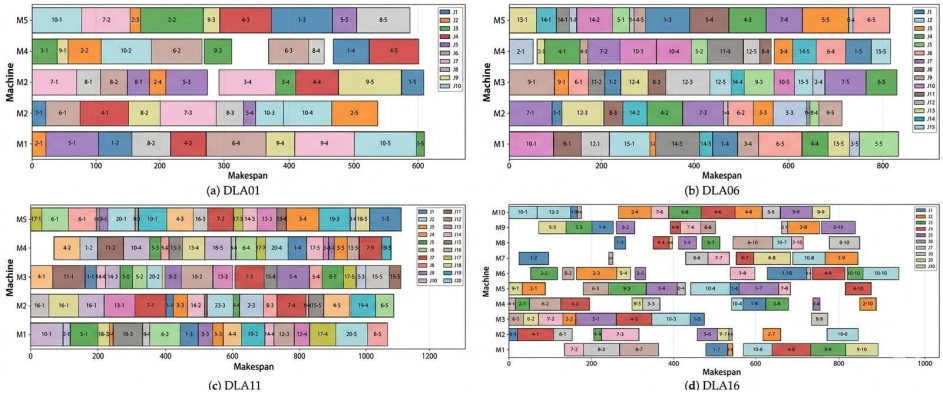


Fig. 9 Scheduling Gantt charts on representative instances

6 Conclusion

This paper investigated the multi-objective distributed flexible job shop scheduling problem (MO-DFJSP) with the objectives of minimizing makespan and total energy consumption. Existing work integrating reinforcement learning with evolutionary algorithms has demonstrated the effectiveness of this paradigm in improving algorithmic convergence, however, reinforcement learning in such work is confined to hyperparameter adjustment, the mutation operator itself still lacks scheduling structure perception capability, and cross-generational population stagnation problems lack hierarchical response mechanisms based on multi-dimensional indicators. Inspired by the above work, this paper proposed HENSGA-DRL, a hybrid NSGA-II algorithm based on disjunctive graph-enhanced deep reinforcement learning, to collaboratively improve search efficiency at both intra-generational gene operation and cross-generational

state regulation levels. Specifically, the CGDRL operator was designed, constructing scheduling constraint topology through disjunctive graphs, extracting bottleneck features via graph convolutional networks, and generating directed machine reassignment decisions through a PPO policy network, so that mutation operations themselves have continuous scheduling structure bottleneck perception and adaptive decision-making capability. Furthermore, the MSHI strategy was proposed, quantifying population health status in real time with hypervolume change rate and objective-space diversity as monitoring indicators, and activating three-tier intervention measures according to the stagnation severity gradient to continuously maintain the vitality of the population evolution trajectory on the cross-generational time scale. Systematic experimental evaluation was conducted on the self-constructed DLA benchmark dataset. Experimental results demonstrate that HENSGA-DRL has significant superiority in solving MO-DFJSP.

Acknowledgements. This work was supported by the Cloud-Edge Collaborative Automobile Manufacturing Lifecycle Management Research (Grant No. SYKL12025B04) and the Hubei Provincial Department of Science and Technology Regional Innovation System Project (Central Government Guided Local Special Fund, Grant No. 2025EIA007).

Declarations

Funding This work was supported by the Cloud-Edge Collaborative Automobile Manufacturing Lifecycle Management Research (Grant No. SYKL12025B04) and the Hubei Provincial Department of Science and Technology Regional Innovation System Project (Central Government Guided Local Special Fund, Grant No. 2025EIA007).

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent for publication Not applicable.

Data availability The DLA benchmark dataset generated during this study is available from the corresponding author on reasonable request.

Materials availability Not applicable.

Code availability The source code is available from the corresponding author on reasonable request.

Author contribution Longyan Xu: Conceptualization, Supervision, Funding acquisition. Li Li: Methodology, Software, Investigation, Writing—original draft. Kai Fang: Validation, Formal analysis. Jinfeng Ye: Data curation, Visualization. NengHui Huang: Writing—review & editing, Resources.

Declarations of generative AI and AI-assisted technologies in the writing process During the preparation of this work, the authors used ChatGPT in order to improve syntax and grammar. After using this tool, the authors reviewed and edited the content as needed and assume full responsibility for the content of the publication.

References

[1] Dauzère-Pérès, S., Ding, J., Shen, L., Tamssaouet, K.: The flexible job shop scheduling problem: A review. European Journal of Operational Research **314**(2), 409–432 (2024) https://doi.org/10.1016/j.ejor.2023.05.017

[2] Para, J., Del Ser, J., Nebro, A.J.: Energy-aware multi-objective job shop scheduling optimization with metaheuristics in manufacturing industries: a critical survey, results, and perspectives. Applied Sciences **12**(3), 1491 (2022) https://doi.org/10.3390/app12031491

[3] Wu, M., Yang, D., Zhou, B., Yang, Z., Liu, T., Li, L., Wang, Z., Hu, K.: Adaptive population NSGA-III with dual control strategy for flexible job shop scheduling problem with the consideration of energy consumption and weight. Machines **9**(12), 344 (2021) https://doi.org/10.3390/machines9120344

[4] Zhang, Z., Fu, Y., Gao, K., Zhang, H., Wang, L.: A cooperative evolutionary algorithm with simulated annealing for integrated scheduling of distributed flexible job shops and distribution. Swarm and Evolutionary Computation **85**, 101467 (2024) https://doi.org/10.1016/j.swevo.2024.101467

[5] Chen, Y., Liao, X., Chen, G., Hou, Y.: Dynamic intelligent scheduling in low-carbon heterogeneous distributed flexible job shops with job insertions and transfers. Sensors **24**(7), 2251 (2024) https://doi.org/10.3390/s24072251

[6] Zhang, W., Xiao, G., Gen, M., *et al.*: Enhancing multi-objective evolutionary algorithms with machine learning for scheduling problems: recent advances and survey. Frontiers in Industrial Engineering **2**, 1337174 (2024) https://doi.org/10.3389/fieng.2024.1337174

[7] Chen, R., Wu, B., Wang, H., Tong, H., Yan, F.: Q-learning based NSGA-II for dynamic flexible job shop scheduling with limited transportation resources. Swarm and Evolutionary Computation **90**, 101658 (2024) https://doi.org/10.1016/j.swevo.2024.101658

[8] Zhu, K., Gong, G., Peng, N., Zhang, L., Huang, D., Luo, Q., Li, X.: Dynamic distributed flexible job-shop scheduling problem considering operation inspection. Expert Systems with Applications **224**, 119840 (2023) https://doi.org/10.1016/j.eswa.2023.119840

[9] Meng, L., Zhang, C., Ren, Y., Zhang, B., Lv, C.: Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem. Computers & Industrial Engineering **142**, 106347 (2020) https://doi.org/10.1016/j.cie.2020.106347

[10] Zhang, G., Hu, Y., Sun, J., Zhang, W.: An improved genetic algorithm for the flexible job shop scheduling problem with multiple time constraints. Swarm and

Evolutionary Computation **54**, 100664 (2020) https://doi.org/10.1016/j.swevo.2020.100664

[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. Computers & Operations Research **132**, 105263 (2021) https://doi.org/10.1016/j.cor.2021.105263

[12] Lin, C.-S., Li, P.-Y., Wei, J.-M., Wu, M.-C.: Integration of process planning and scheduling for distributed flexible job shops. Computers and Operations Research **124**, 105053 (2020) https://doi.org/10.1016/j.cor.2020.105053

[13] Xie, J., Li, X., Gao, L., Gui, L.: A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems. Journal of Manufacturing Systems **71**, 82–94 (2023) https://doi.org/10.1016/j.jmsy.2023.08.002

[14] Li, X., Xie, J., Ma, Q., Gao, L., Li, P.: Improved gray wolf optimizer for distributed flexible job shop scheduling problem. Science China Technological Sciences **65**, 2105–2115 (2022) https://doi.org/10.1007/s11431-022-2096-6

[15] Cao, S., Li, R., Gong, W., Lu, C.: Inverse model and adaptive neighborhood search based cooperative optimizer for energy-efficient distributed flexible job shop scheduling. Swarm and Evolutionary Computation **83**, 101419 (2023) https://doi.org/10.1016/j.swevo.2023.101419

[16] Luo, Q., Deng, Q., Gong, G., Guo, X., Liu, X.: A distributed flexible job shop scheduling problem considering worker arrangement using an improved memetic algorithm. Expert Systems with Applications **207**, 117984 (2022) https://doi.org/10.1016/j.eswa.2022.117984

[17] Zhang, Z.-Q., Li, Y., Qian, B., Hu, R., Yang, J.-B.: A multidimensional probabilistic model based evolutionary algorithm for the energy-efficient distributed flexible job-shop scheduling problem. Engineering Applications of Artificial Intelligence **135**, 108841 (2024) https://doi.org/10.1016/j.engappai.2024.108841

[18] Gu, Y., Xu, H., Yang, J., Li, R.: An improved memetic algorithm to solve the energy-efficient distributed flexible job shop scheduling problem with transportation and start-stop constraints. Mathematical Biosciences and Engineering **20**, 21467–21498 (2023) https://doi.org/10.3934/mbe.2023950

[19] Li, R., Gong, W., Wang, L., Lu, C., Zhuang, X.: Surprisingly popular-based adaptive memetic algorithm for energy-efficient distributed flexible job shop scheduling. IEEE Transactions on Cybernetics **53**, 8013–8023 (2023) https://doi.org/10.1109/TCYB.2022.3164830

[20] Du, B., Han, S., Guo, J., Li, Y.: A hybrid estimation of distribution algorithm for solving assembly flexible job shop scheduling in a distributed environment.

Engineering Applications of Artificial Intelligence **133**, 108491 (2024) https://doi.org/10.1016/j.engappai.2024.108491

[21] Wang, J., Tang, H., Lei, D.: A Q-learning artificial bee colony for distributed assembly flow shop scheduling with factory eligibility, transportation capacity and setup time. Engineering Applications of Artificial Intelligence **123**, 106230 (2023) https://doi.org/10.1016/j.engappai.2023.106230

[22] Tian, S., Zhang, C., Fan, J., Gu, X., Liu, Q.: A genetic algorithm with critical path-based variable neighborhood search for distributed assembly job shop scheduling problem. Swarm and Evolutionary Computation **85**, 101485 (2024) https://doi.org/10.1016/j.swevo.2024.101485

[23] He, C., Li, L., Tian, Y., Zhang, X., Cheng, R., Jin, Y., Yao, X.: Accelerating large-scale multiobjective optimization via problem reformulation. IEEE Transactions on Evolutionary Computation **23**, 949–961 (2019) https://doi.org/10.1109/TEVC.2019.2896068

[24] Chen, R., Yang, B., Li, S., Wang, S.: A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. Computers and Industrial Engineering **149**, 106778 (2020) https://doi.org/10.1016/j.cie.2020.106778

[25] Pan, Z., Lei, D., Wang, L.: A bi-population evolutionary algorithm with feedback for energy-efficient fuzzy flexible job shop scheduling. IEEE Transactions on Systems Man and Cybernetics: Systems **52**, 5295–5307 (2022) https://doi.org/10.1109/TSMC.2021.3120702

[26] Yu, H., Gao, K., Li, Z., Suganthan, P.N.: Energy-efficient multi-objective distributed assembly permutation flowshop scheduling by Q-learning based metaheuristics. Applied Soft Computing **166**, 112247 (2024) https://doi.org/10.1016/j.asoc.2024.112247

[27] Li, Y., Chen, J., Duan, X., Pan, R., He, Z.: Knowledge-based reinforcement learning and estimation of distribution algorithm for flexible job shop scheduling problem. IEEE Transactions on Emerging Topics in Computational Intelligence **7**, 1036–1050 (2023) https://doi.org/10.1109/TETCI.2022.3145706

[28] Li, R., Gong, W., Lu, C., Wang, L.: A learning-based memetic algorithm for energy-efficient flexible job-shop scheduling with type-2 fuzzy processing time. IEEE Transactions on Evolutionary Computation **27**, 610–620 (2023) https://doi.org/10.1109/TEVC.2022.3175832

[29] Zhang, C., Song, W., Cao, Z., Zhang, J., Tan, P.S., Chi, X.: Learning to dispatch for job shop scheduling via deep reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1621–1632 (2020)

[30] Song, W., Chen, X., Li, Q., Cao, Z.: Flexible job-shop scheduling via graph neural network and deep reinforcement learning. IEEE Transactions on Industrial Informatics **19**, 1600–1610 (2023) https://doi.org/10.1109/TII.2022.3189725

[31] Lei, K., Guo, P., Zhao, W., Wang, Y., Qian, L., Meng, X., Tang, L.: A multi-action deep reinforcement learning framework for flexible job-shop scheduling problem. Expert Systems with Applications **205**, 117796 (2022) https://doi.org/10.1016/j.eswa.2022.117796

[32] Tao, X.-R., Pan, Q.-K., Sang, H.-Y., Duan, P.-Y., Gao, L.: Nondominated sorting genetic algorithm-II with Q-learning for the distributed permutation flowshop rescheduling problem. Knowledge-Based Systems **278**, 110880 (2023) https://doi.org/10.1016/j.knosys.2023.110880

[33] Zhang, Z.-Q., Wu, F.-C., Qian, B., Hu, R., Liu, B., Chang, Y.-L.: A Q-learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation. Expert Systems with Applications **234**, 121050 (2023) https://doi.org/10.1016/j.eswa.2023.121050

[34] Zhang, Z., Shao, Z., Shao, W., Pi, D., Wu, C.: MRLM: a meta-reinforcement learning-based metaheuristic for hybrid flowshop scheduling problem with learning and forgetting effects. Swarm and Evolutionary Computation **85**, 101479 (2024) https://doi.org/10.1016/j.swevo.2024.101479

[35] Zhang, Z., Tang, Q., Zhang, L., Li, Z., He, Z.: A Q-learning-based multi-population algorithm for multi-objective distributed heterogeneous assembly no-idle flowshop scheduling with batch delivery. Expert Systems with Applications **263**, 125690 (2025) https://doi.org/10.1016/j.eswa.2024.125690

[36] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002) https://doi.org/10.1109/4235.996017

[37] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

[38] Bierwirth, C.: A generalized permutation approach to job shop scheduling with genetic algorithms. OR Spectrum **17**(2–3), 87–92 (1995) https://doi.org/10.1007/BF01719250

[39] Xiao, B., Zhao, Z., Wu, Y., Zhu, X., Peng, S., Su, H.: An improved MOEA/D for multi-objective flexible job shop scheduling by considering efficiency and cost. Computers & Operations Research **167**, 106674 (2024) https://doi.org/10.1016/j.cor.2024.106674

[40] Wu, X., Peng, J., Xiao, X., Wu, S.: Many-objective flexible job shop scheduling problem with green consideration. Energies **15**(5), 1884 (2022) https://doi.org/

10.3390/en15051884

[41] Sang, Y., Tan, J.: Intelligent factory many-objective distributed flexible job shop collaborative scheduling method. Computers and Industrial Engineering **164**, 107884 (2022) https://doi.org/10.1016/j.cie.2022.107884