Solving Heterogeneous Distributed Job-Shop Scheduling Problem (DFJSP) based on deep reinforcement learning

① Co-Evolution With DRL for Energy-Aware Heterogeneous DFJSP

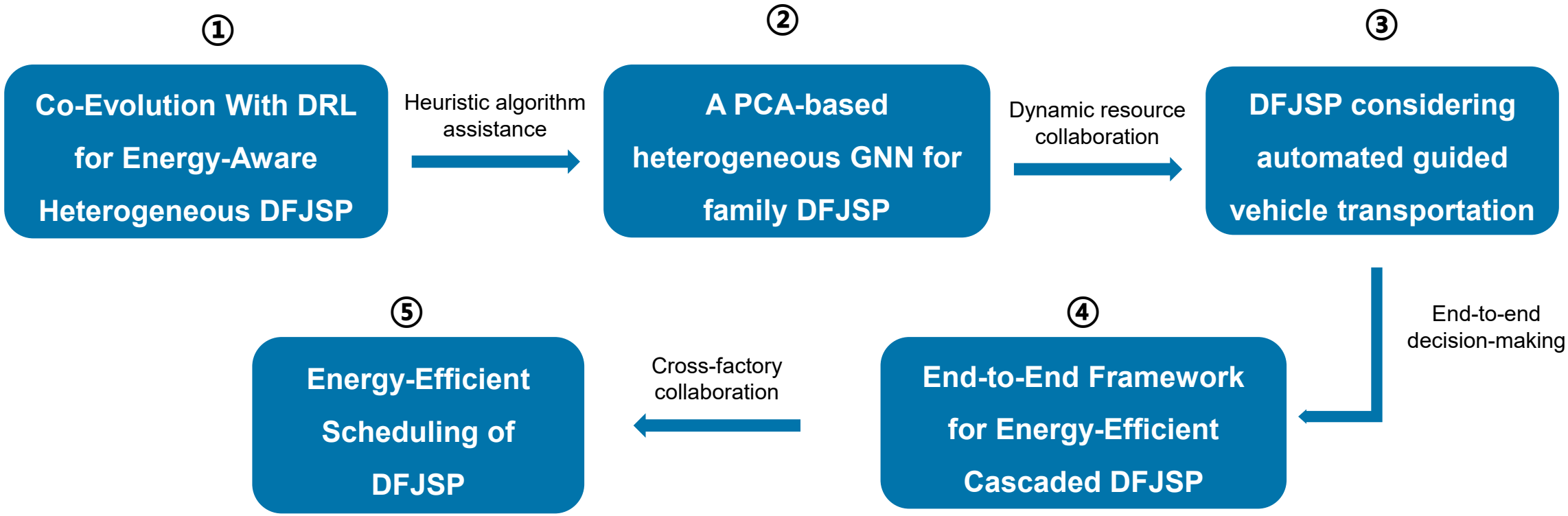
② A PCA-based heterogeneous GNN for a family DFJSP

③ DFJSP considering automated guided vehicle transportation

④ End-to-End Framework for Energy-Efficient Cascaded DFJSP

⑤ Energy-Efficient Scheduling of DFJSP

Solving Heterogeneous Distributed Job-Shop Scheduling Problem (DFJSP) based on deep reinforcement learning



Paper 1

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

Co-Evolution With Deep Reinforcement Learning for Energy-Aware Distributed Heterogeneous Flexible Job Shop Scheduling

Rui Li , Wenyin Gong, Ling Wang, Chao Lu , and Chenxin Dong

Publish Journal: IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS

Published Time: January 2024

IF: 8.7

Contents

Contents

Problem Description

Energy-aware distributed heterogeneous flexible job shop scheduling (DHFJS) problem is an extension of the traditional FJS, which is harder to solve. This work aims to minimize total energy consumption (TEC) and makespan for DHFJS.

A deep Q-networks-based co-evolution algorithm (DQCE) is proposed to solve this NP-hard problem, which includes four parts: First, a new co-evolutionary framework is proposed, which allocates sufficient computation to global searching and executes local search surrounding elite solutions.

Nine problem features-based local search operators are designed to accelerate convergence. Moreover, deep Q-networks are applied to learn and select the best operator for each solution. Furthermore, an efficient heuristic method is proposed to reduce TEC.

Contents

Method Design

Problem Description

The DHFJS should solve the following three subproblems: 1) *Factory Assignment (FA):* determining the processing factory for all jobs; 2) *Operation Sequencing:* arranging a reasonable sequence for all operations assigned to each factory; and 3) *Machine Selection (MS):* selecting a suitable machine for each operation from flexible machines set. DHFJS includes n_f heterogeneous factories and n jobs dispatched to different factories, where every job has n_i operations. Moreover, the total operations of a job must be assigned to the same factory. As for the same operation $O_{i,j}$, it has the same selectable machine set but different processing times in different factories.

Method Design

Optimization objective

1) Makespan Decision Expression: Makespan is the maximum completion time of scheduling, representing an enterprise's production profit. Moreover, the makespan $C_{\max}$ objective is described

$$\min \ F_1 = C_{\max} = \max\left\{F_{f,k,t}\right\} \forall f \in F, k \in H_f, t \in P_{f,k}. \quad (1)$$

2) TEC Decision Expression: TEC during processing in all factories is regarded as a critical green indicator, representing a factory's carbon emissions. TEC is stated as follows:

$$\min \ F_2 = \mathrm{TEC} = G_P + G_I \quad (2)$$

$$G_P = \sum_{i \in I} \sum_{j \in J_i} \sum_{f \in F} \sum_{k \in M} \sum_{t \in P_{f,k}} \mathbf{X}_{i,j,f,k,t} \cdot T_{i,j,f,k} \cdot W_O \quad (3)$$

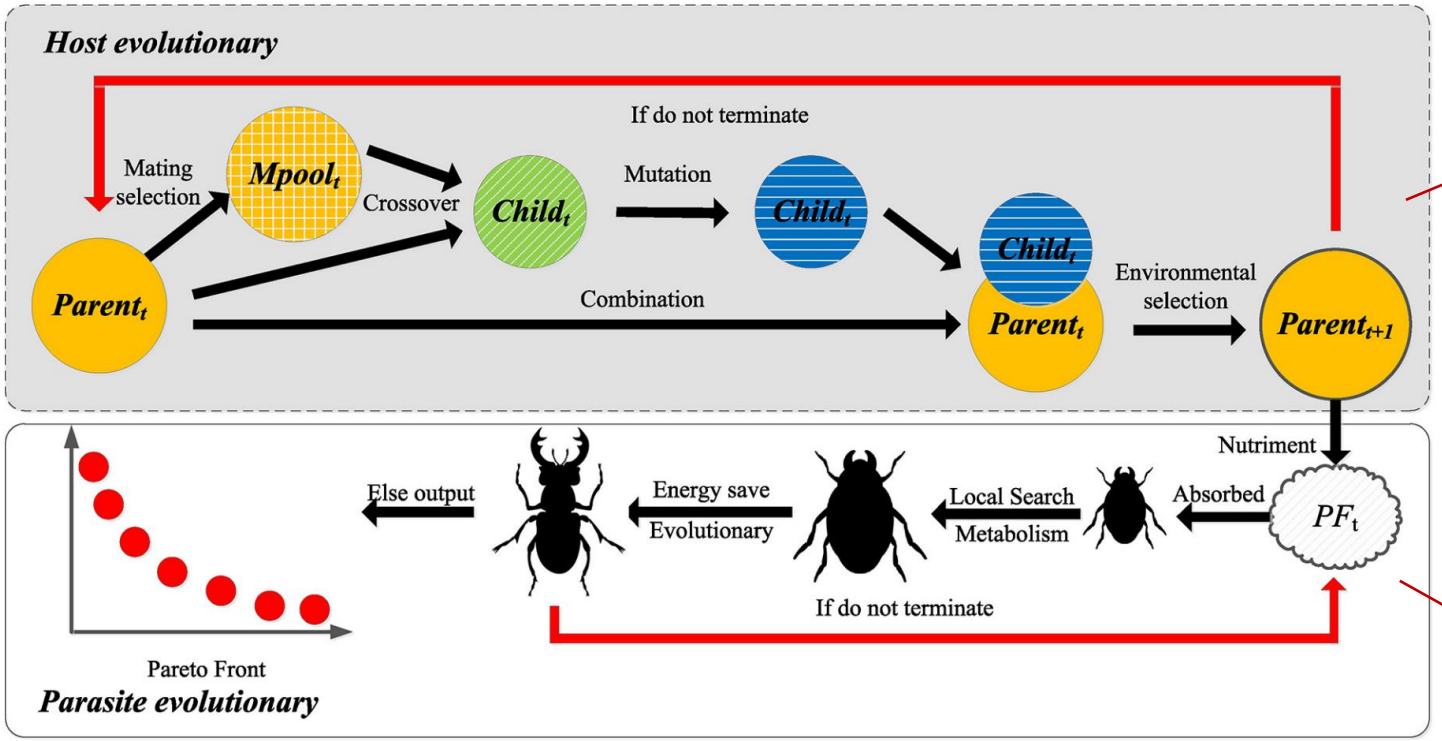
$$G_I = \sum_{f \in F} \sum_{k \in K} \sum_{t \in P'_{f,k}} W_I \cdot \left(B_{f,k,t+1} - F_{f,k,t}\right). \quad (4)$$

Method Design

Motivations of DQCE

1) The number of combinations of DHFJS is $(\sum n_i)! \cdot m! \cdot n_f!$, which is too hard to solve. Under limited computation resources, the global search can generate a considerable perturbation, which can approach potential objective space. Thus, more computational resources should be allocated to the global search than the local search. Therefore, the global and local searches are put into two types of populations.
2) Since DHFJS is first studied, the previous works have not designed an efficient local search for this problem. Thus, nine neighborhood structures based on problem features are designed to accelerate algorithmic convergence.
3) Due to its ability to rapidly learn the mapping between solutions and operators, DQN is applied to learn and select the best local search operator for each individual.
4) The previous works cannot provide an efficient energy-saving strategy. Based on observing the TEC objective, idle time is the key variable influencing TEC. Finding the idle space and inserting the latter operation into it can reduce TEC. Meanwhile, the former operation can be backward inserted into the latter idle space, thus further reducing the TEC.

Co-Evolutionary Framework Based on Parasitic Behavior



Host: Global search

Host: Local optimization

Fig. 1. PCE for distributed shop scheduling.

Encoding and Decoding Schema

② Process determination

③ Machine selection

① Factory selection

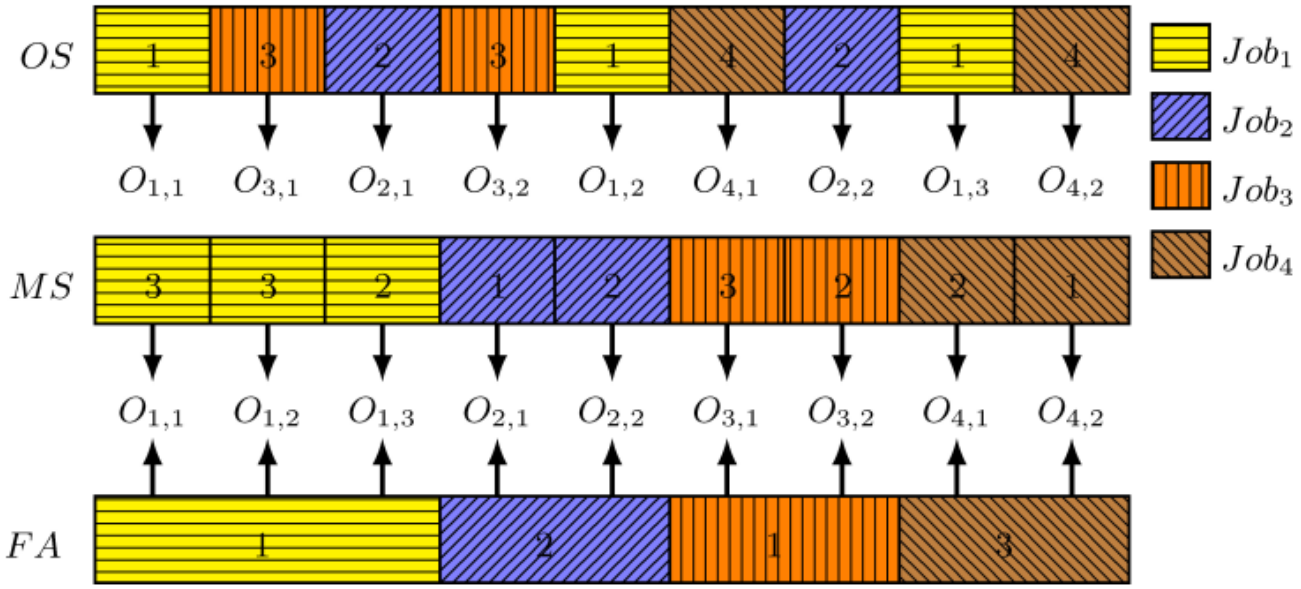


Fig. 2. Solution representation for DHFJS.

Knowledge-Driven Local Search Strategies

Consecutive processes on the same machine

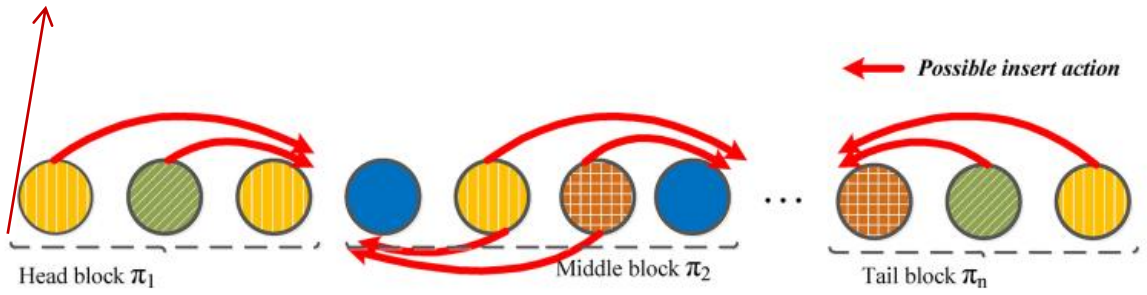


Fig. 3. N6 neighborhood structure. Different operations of a job have the same color. For example, the yellow operations belongA job I_1. $O_{1,1}$ and $O_{1,2}$ are on the same machine, and $O_{1,3}$ is on another machine.

1) $\mathcal{N}_1$ (N6): N6 is a strong neighborhood structure designed for job shop scheduling problems by changing the OS in critical blocks. A block comprises several adjacent critical operations in the whole path on the same machine [44]. Moreover, N6 executes the following steps to reduce makespan. As shown in Fig. 3, for the head block π_1 in the critical path, an operation except for the tail operation is randomly selected and is inserted into the tail of π_1. For the middle block π_2, two middle operations are randomly selected, one operation is inserted into the head of π_2, and another is inserted into the tail of π_2. Finally, for the tail block π_n, an operation except for the head operation is randomly inserted into the front of π_n.

Deep-Q-Network-Based Strategies Selection Model

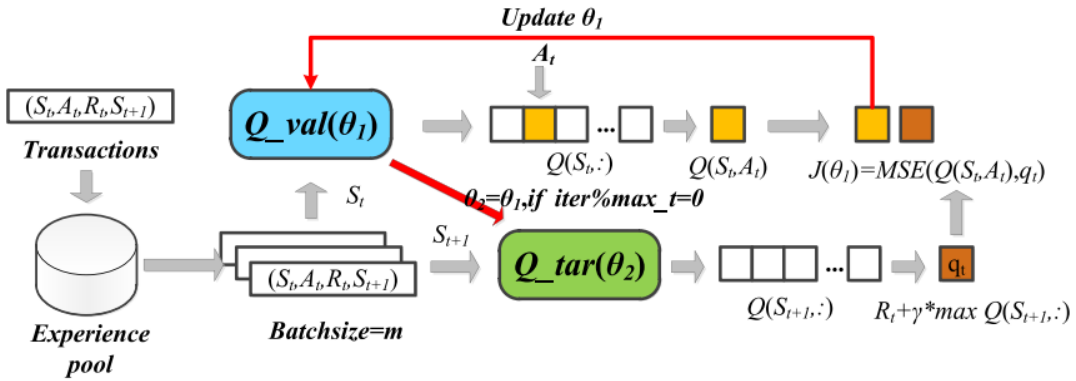


Fig. 4. Training process of DQN.

State Definition: To better learn the solution distribution in DHFJS, the state is a vector merged by (OS, MS, FA). Moreover, the length of OS and MS equals the number of total operations $\sum_{i=1}^{n} n_i$, and the length of FA equals the number of jobs n.

Action and Reward Definitions: This work applies the DQN to tailor a neighborhood structure for each solution in the parasite population which can significantly improve the success rate of finding a new nondominated solution. Thus, the actions in DQN are the nine neighborhood structures mentioned in Section III-F. The reward feedback is set as follows: if the new solution replaces the old one, then the reward is five; Else if the new and old solutions cannot dominate each other, then the reward is ten; Else the reward is zero. The reward value is usually set to ten when the selected action succeeds and set to zero when it fails according to [46]. Ten is an appropriate reward. If the reward is set too large, it will increase the difficulty of DQN convergence, and the loss function will fluctuate. If the reward is set too small, it is difficult to distinguish the feedback of successful and failed actions. In order to reduce learning difficulty, no punishment is added to DQN and the reward is set to zero when the action fails. In order to guide the DQN to find more nondominated solutions, the reward is set to ten when the action finds a nondominated solution. Meanwhile, the reward is smaller when action finds a new solution, replacing the old.

Energy-Saving Strategy

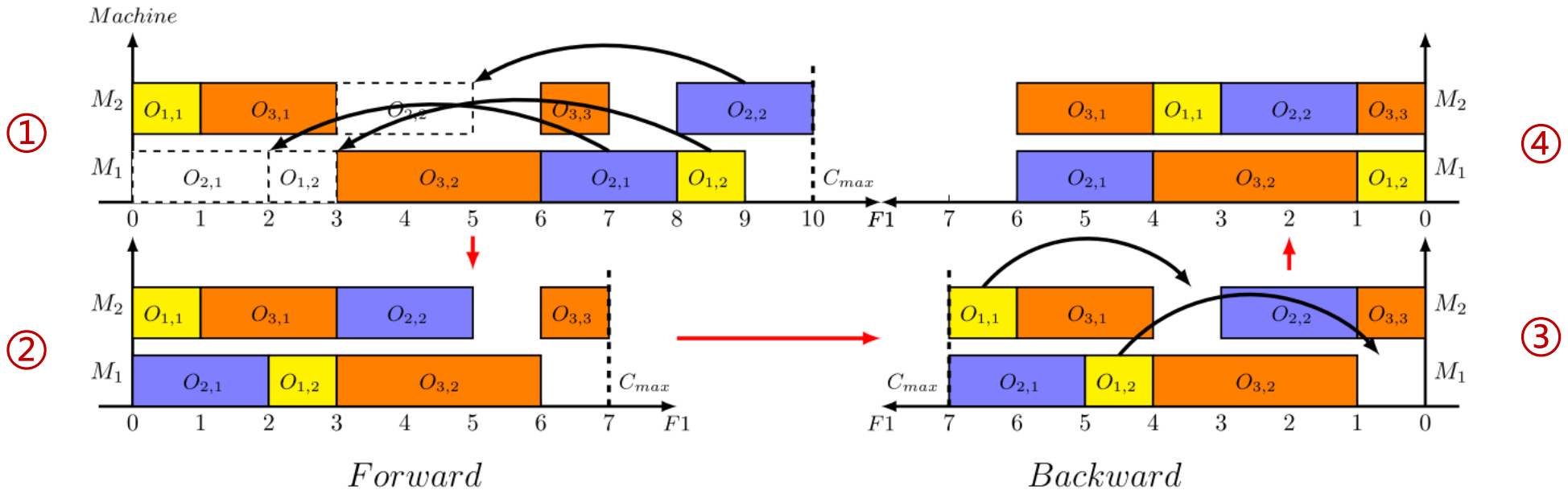


Fig. 5. Energy-saving strategy.

Contents

Experiment Results

Dataset

Since DHFJS is not studied before, various scales of instances are generated to evaluate the performance of DQCE. The number of jobs is $n \in \{10, 20, 30, 40, 50\,100\,150, 200\}$, and the number of factories ranges from $n_f \in \{2, 3, 4, 5, 6, 7\}$. The number of machines in each heterogeneous factory is $m = 5$, and the number of operations of each job is $n_i = 5$. Every operation's processing time in each factory is $P_{f,i,j,k} \in [5, 20]$. The processing and idle powers are chosen as $W^O_{i,j,f,k} = 4.0$ kWh and $W_I = 1.0$ kWh, respectively. Finally, 20 instances with various scales are generated. The stop criterion is MaxNEFs $= 200 \times \sum_1^n n_i$.

Dataset

Hypervolume (HV) [47], generation distance (GD), and Spread [41] are utilized to evaluate comprehensive performance, convergence, and diversity of multiobjective evolutionary algorithms (MOEAs). Furthermore, when the GD or Spread values are lower, the convergence and diversity of an algorithm are better. Whereas, If the HV value is bigger, the comprehensive performance of an algorithm is better.

Experiment Results——Parameters Calibration

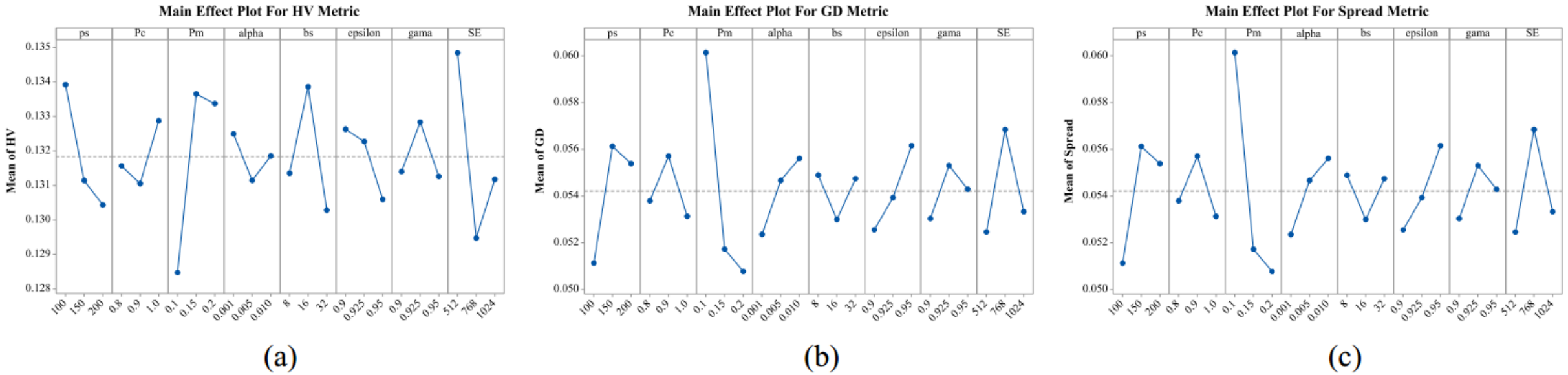


Fig. 6. Three metrics' main effect plots. (a) HV. (b) GD. (c) Spread.

Experiment Results——Ablation Experiment

TABLE I
RANK RESULTS OF THE FRIEDMAN RANK-AND-SUM TEST OF ALL VARIANT ALGORITHMS (SIGNIFICANT LEVEL $\alpha = 0.05$)

MOEAs	HV		GD		Spread	
	rank	p-value	rank	p-value	rank	p-value
MA	5.00		5.00		3.75	
PCE-NS	3.45		3.75		3.75	
PCE-ES	2.95	1.49E-29	2.75	3.34E-29	2.20	6.80E-04
PCE	2.60		2.55		3.10	
DQCE	**1.00**		**1.00**		**2.20**	

Experiment Results—— Comparison and Discussions

TABLE II
RANK RESULTS OF THE FRIEDMAN RANK-AND-SUM TEST OF ALL COMPARISON ALGORITHMS (SIGNIFICANT LEVEL $\alpha = 0.05$)

MOEAs	HV		GD		Spread	
	rank	p-value	rank	p-value	rank	p-value
NSGA-II	5.75		5.25		3.50	
MOEA/D	2.80		3.00		5.40	
TS-NSGA-II	4.30		3.80		3.25	
HSLFA	4.55	2.10E-19	5.00	9.14E-19	4.65	8.26E-08
LRVMA	2.65		2.95		5.35	
MOEA/D-DQN	6.95		7.00		3.95	
DQCE	**1.00**		**1.00**		**1.90**	

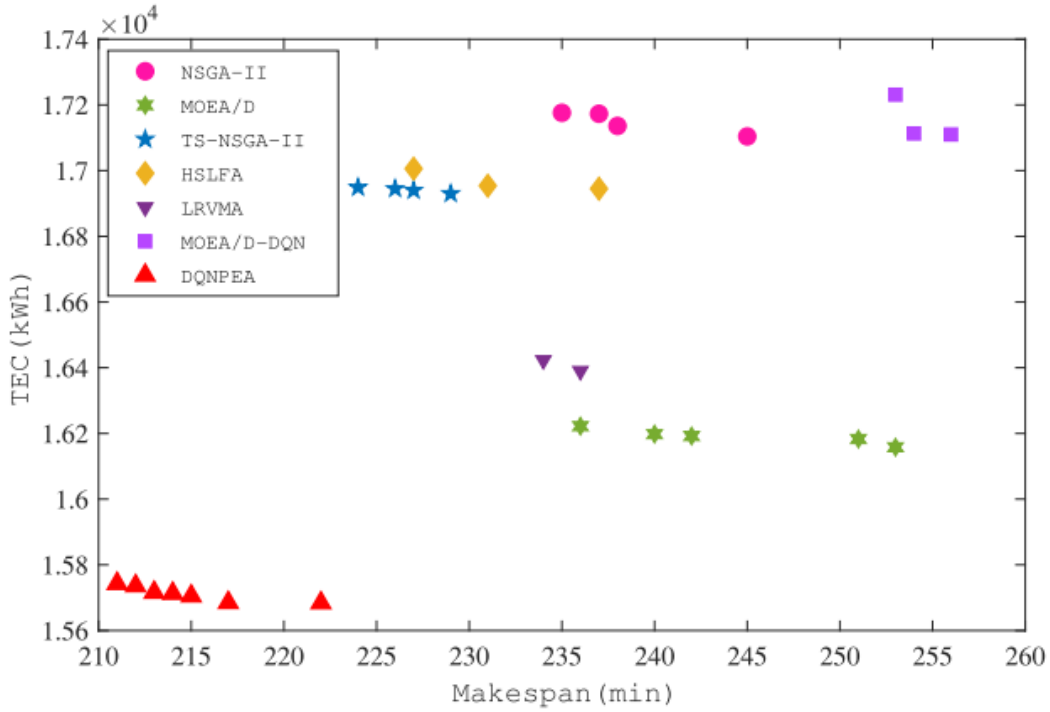


Fig. 7. Approximate Pareto Front found by each algorithm on 100J4F.

Experiment Results—— Comparison and Discussions

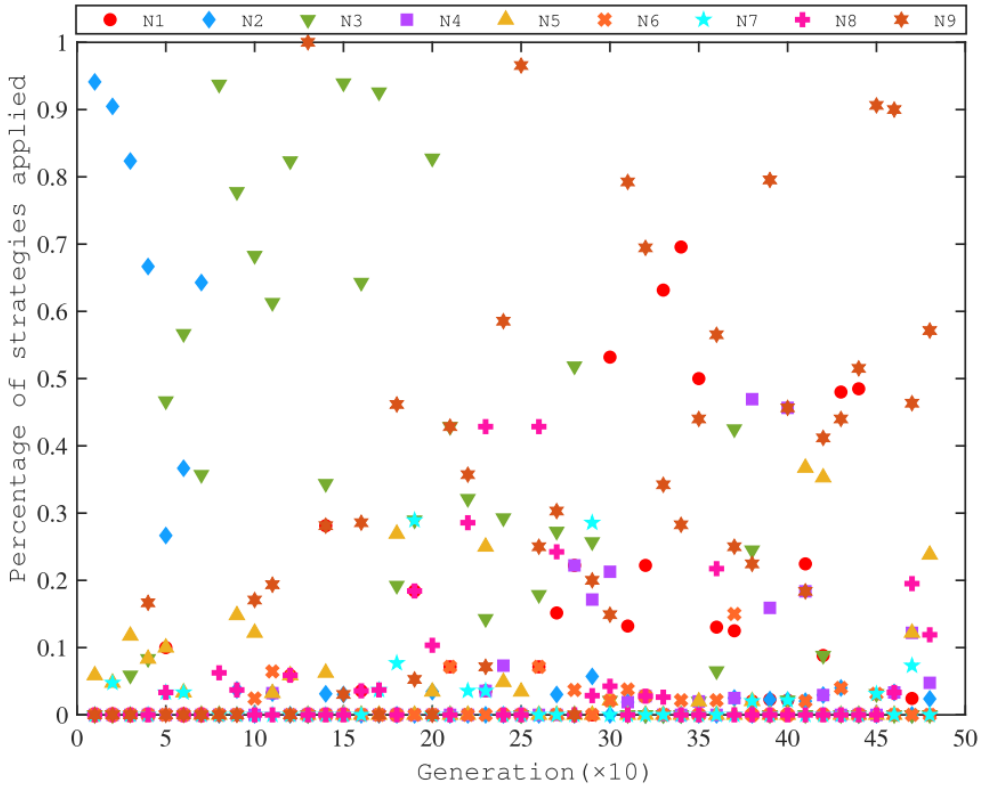


Fig. 8. Ratio of strategies selected by DQCE during the optimization process of solving 100J4F.

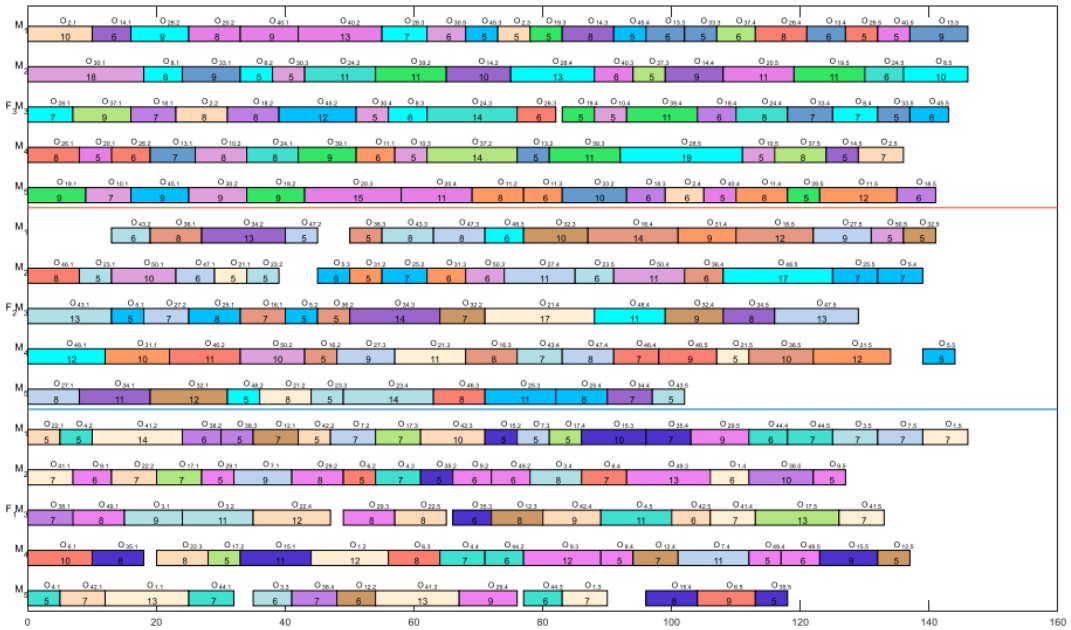


Fig. 9. Gantt chart of solution A with the best C_{max} on 50J3F.

Experiment Results—— Comparison and Discussions

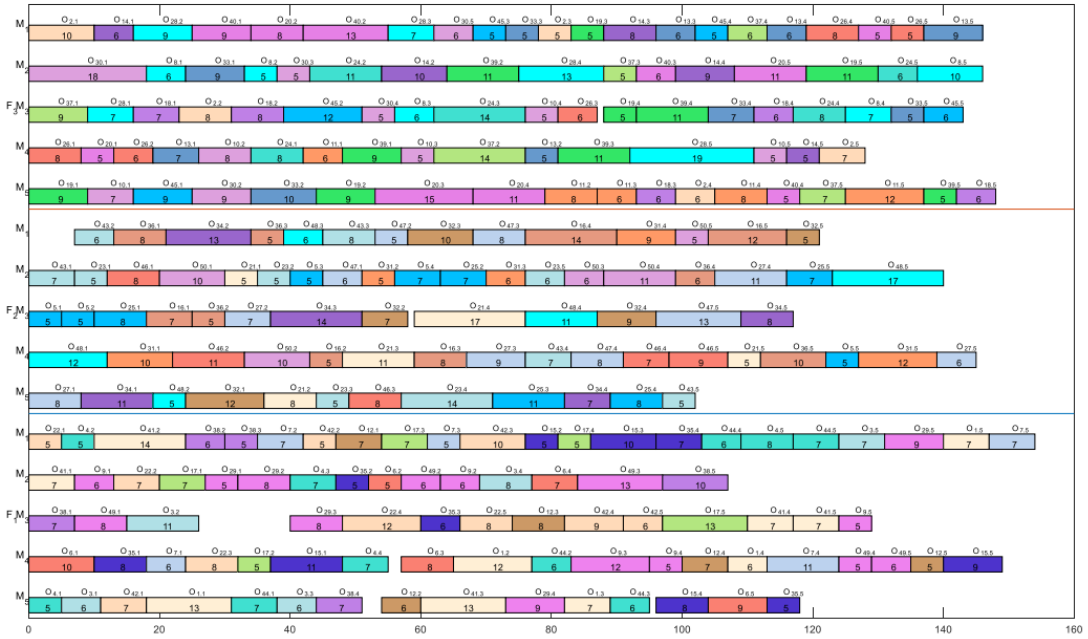


Fig. 10. Gantt chart of solution A with the best TEC on 50J3F.

TABLE III
STATISTICAL RESULTS AMONG THREE METRICS OF ALL COMPARISON ALGORITHMS IN REAL-WORLD CASE

MOEAs	HV		GD		Spread	
	mean	std	mean	std	mean	std
NSGA-II	0.167102	0.000574	0.001878	0.000917	0.943469	0.110437
MOEA/D	0.165206	0.000996	0.002287	0.00066	1.019607	**0.101734**
TS-NSGA-II	0.16741	0.000433	0.001892	0.000725	0.973863	0.109112
HSLFA	0.164324	0.001245	0.003141	0.001154	1.112472	0.309525
LRVMA	0.164192	0.001149	0.003522	0.00168	1.150408	0.327005
MOEA/D-DQN	0.166126	0.000549	0.001716	0.000726	1.090645	0.111967
DQCE	**0.167491**	**0.000287**	**0.001497**	**0.000401**	**0.929345**	0.10818
p-value	8.60E-19		1.62E-07		5.78E-05	

Contents

Future Work

①

1) enhancing the explanation of the reasoning process of DQN;

2) considering end-to-end network for DHFJS;

3) considering the differential distributed heterogeneous FJSP with different machine numbers, machine failure, or maintenance.

Paper 2

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

HGNP: A PCA-based heterogeneous graph neural network for a family distributed flexible job shop

Jiake Li, Junqing Li, Ying Xu

Publish Journal: Computers & Industrial Engineering

Published Time: December 31, 2024

IF: 6.5

Contents

Contents

Problem Description

In this study, first, a mixed integer programming (MIP) model is formulated for the DFJSP with family setup time. To minimize the makespan, a hybrid heterogeneous graph neural network with a principal component analysis (PCA)-based transform mechanism (HGNP) is proposed.

In the proposed algorithm, a novel state representation is designed, which combines the features of operation, machine and factory assignment. Then, a multilayer perceptron (MLP) mechanism is used for the operation embedding, and graph attention networks (GATs) are embedded for the machine and factory embeddings. Next, a PCA-based transform mechanism is developed to further fuse all the three embeddings.

Contents

Method Design

Distributed FJSP definition

A DFJSP is a typical scheduling problem, that has the following features. First, there are n jobs to be processed on m machines in h factories. Each job has a certain number of operations, and the processing routine of each job is independent of each other. There are parallel machines for each operation, and therefore, one task is to select a suitable machine for each operation. There are parallel factories that should be selected for each job as well. After assigning operations on machines in factories, we should decide the processing sequences.

Method Design

Decision variables

- $Y_{i,j,k,r,f}$: a binary value that is set to 1 if $O_{i,j}$ is assigned to the rth position on machine k in factory f; otherwise, it is set to 0.
- $Z_{i,j,f}$: a binary value that is set to 1 if $O_{i,j}$ is assigned to factory f; otherwise, it is set to 0.

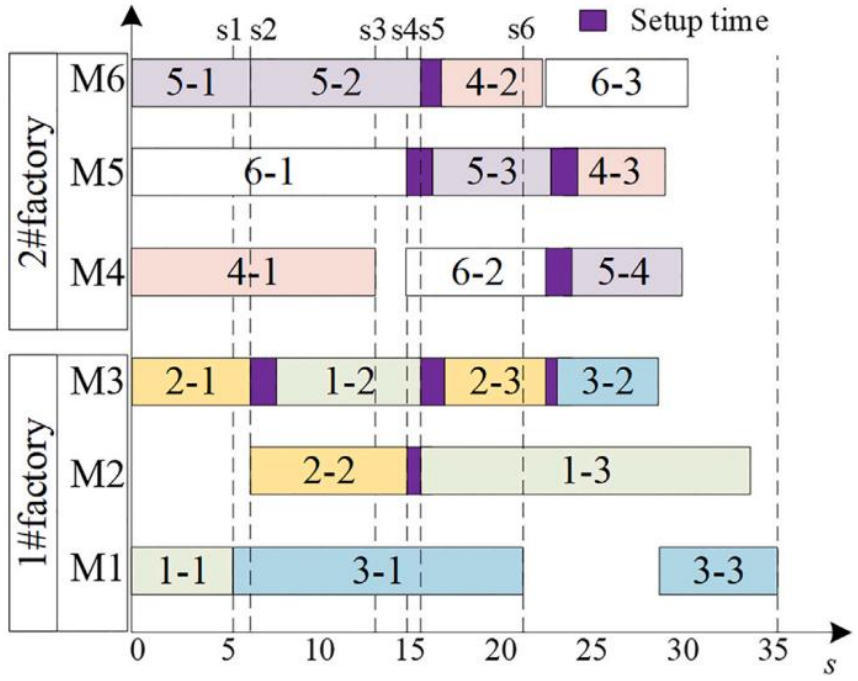


Fig. 2. Illustration of a complete solution with states.

Objective of the considered problem

The objective of this study is to minimize the maximum completion time of all operations. Eq. (2) lists the calculation of the objective.

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

$$C_{\max} = \max\{C_{i,j,f}\}, \forall j, q, f \tag{2}$$

Family setup time

The family/group setup time constraints illustrated in Constraints (17) and (18) ensure that the setup time between two jobs should be considered, if the following conditions are satisfied: (a) the two jobs are processed on consecutive positions of the same machine; and (b) the two jobs are not in the same family or group.

$$\sum_{i \neq i'} \sum_{j} ST_{i,j,i',j'} \times Y_{i,j,k,r,f} \leq S_{k,r+1,f} - C_{k,r,f} + L \times$$

$$(1 - Y_{i',j',k,r+1,f}), \forall i', k \in M_{i',j'}, j', f, r, G_i \neq G_i' \tag{17}$$

$$\sum_{i \neq i'} \sum_{j} ST_{i,j,i',j'} \times Y_{i,j,k,r,f} + L \times (1 - Y_{i',j',k,r+1,f}) \geq$$

$$S_{k,r+1,f} - C_{k,r,f}, \forall i', k \in M_{i',j'}, j', f, r, G_i \neq G_i' \tag{18}$$

Method Design

Markov Decision Process (MDP)

States

The features of the machines contain four parts: (1) τ^1_{mf}: number of neighboring operations, which indicates the number of operations that can be processed by the current machine. (2) τ^2_{mf}: machine available time. (3) τ^3_{mf}: machine utilization, which is used to calculate the ratio between the total processing times of the current machine, and the current time t. (4) τ^4_{mf}: current processing operation, which is used to calculate the setup time between two operations processed on the same machine.

The features of the factories contain two parts: (1) τ^1_f: number of assigned jobs at the current factory. (2) τ^2_f: factory utilization, which is used to calculate the ratio between the total processing time of the current factory and the current time t.

The features of the O-M arcs tell the relations between operations and machines, which contain only one part, i.e., τ^1_{om}, a binary value set to "1" if the operation is assigned to the corresponding machine. The features of the J-F arcs tell the relations between jobs and factories, which contain only one part, i.e., τ^1_{jf}, a binary value set to "1" if the job is assigned to the corresponding factory.

Altogether, the state representation s_t is given by

$$s_t = \{\tau^1_{op}, \ldots, \tau^6_{op}; \tau^1_{mf}, \ldots, \tau^4_{mf}; \tau^1_f, \tau^2_f; \tau^1_{om}; \tau^1_{jf}\} \quad (19)$$

Action

- Factory select
- Operation select
- Machine allocation

Reward

Makespan (Job Maximum completion time)

Embedding mechanism

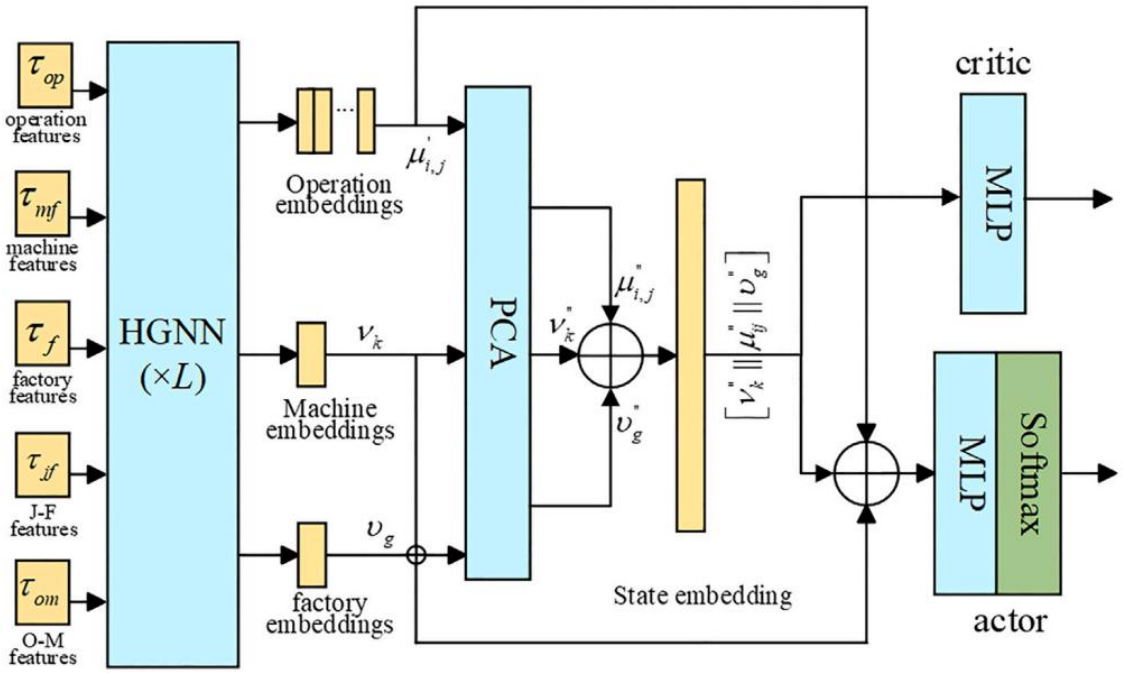


Fig. 1. Structure of the proposed HGNP.

Machine node embedding (GAT)

Step 1. Given a machine M_k, for all the neighboring operations in $O_{i,j} \in N_t(M_k)$, perform the following two steps.

Step 2. Concatenate τ_{op}^{ij} and τ_{om}^{ijk} to construct the resulting feature $\mu_{ijk} = [\tau_{op}^{ij} \parallel \tau_{om}^{ijk}] \in R^7$.

Step 3. Calculate the attention coefficients on the edge between M_k and $O_{i,j}$ as follows.

$$e_{ijk} = \text{LeakyReLu}\left(a^{\mathrm{T}}[W^M \tau_{mf}^k \parallel W^O \mu_{ijk}]\right) \tag{24}$$

where, $W^M \in R^{d\times 4}$, $W^O \in R^{d\times 7}$, $\tau_{mf}^k \in R^{d\times 4}$, and $a^T \in R^{2d}$.

Step 4. Calculate the attention coefficients on M_k itself as follow.

$$e_{kk} = \text{LeakyReLu}\left(a^{\mathrm{T}}[W^M \tau_{mf}^k \parallel W^M \tau_{mf}^k]\right) \tag{25}$$

Step 5. Calculate the normalized attention coefficients as follows.

$$\alpha_{ijk} = \sigma\left(Norm\left([e_{ijk} \parallel e_{kk}]\right)\right)_{1:N} \tag{26}$$

$$\alpha_{kk} = \sigma\left(Norm\left([e_{ijk} \parallel e_{kk}]\right)\right)_{N+1} \tag{27}$$

where, $\sigma(.)$ is the softmax function, $Norm(.)$ is the normalize function, $(.)_{1:N}$ is to obtain the first N lines, and $(.)_{N+1}$ is to obtain the $(N+1)^{th}$ line.

Step 6. The machine embedding v_k for M_k is calculated as follows.

$$v_k = \sigma\left(\alpha_{kk} W^M \tau_{mf}^k + \sum_{O_{i,j}\in N_t(M_k)} \alpha_{ijk} W^O \mu_{ijk}\right) \tag{28}$$

Embedding mechanism

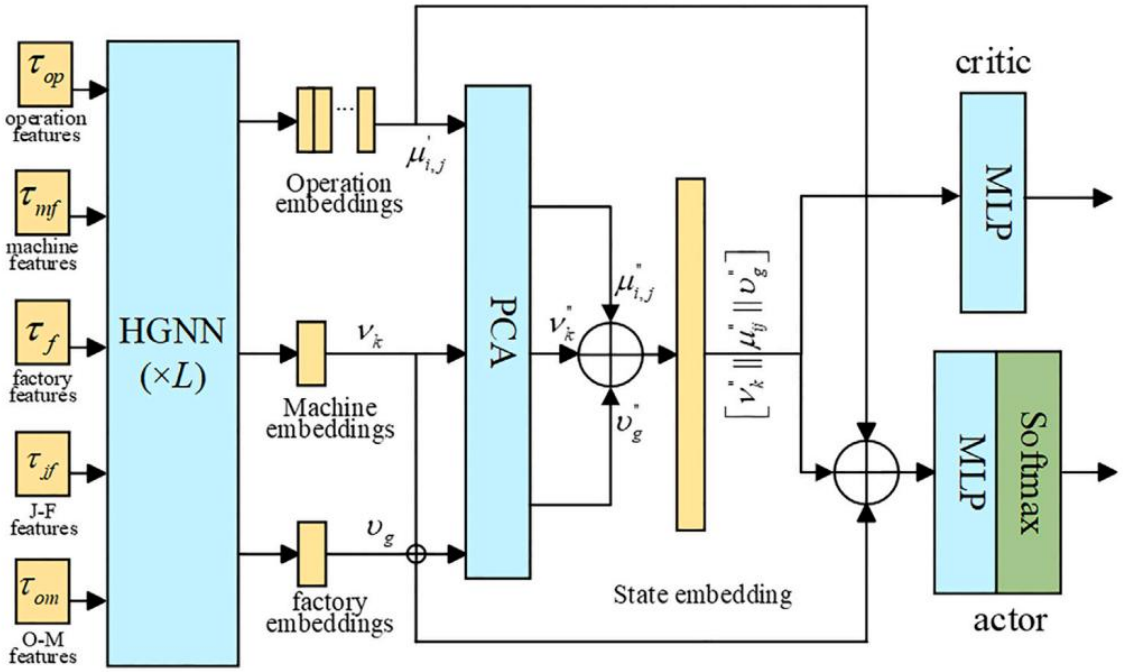


Fig. 1. Structure of the proposed HGNP.

Operation node embedding (MLP)

The multilayer perceptron (MLP) mechanism is used for the operation node embedding $\mu'_{i,j}$, in which there are two d_h-dimensional hidden layers, d-dimensional output, and ELU as the activation function. Four MLP layers are used to capture four types of information, i.e., the neighboring machines of $O_{i,j}$ ($v_{ij} = \sum_{k \in N_t(O_{i,j})} \tau^k_{mf}$), $O_{i,j-1}(\tau^{i,j-1}_{op})$, $O_{i,j}$, and $O_{i,j+1}$.

The following formulation is used to combine the four features to construct the operation node embedding.

$$\mu'_{i,j} = \mathrm{MLP}_{\theta_0}\left(\begin{array}{l} \mathrm{ELU}[\mathrm{MLP}_{\theta_1}(v_{ij}) \,\|\, \mathrm{MLP}_{\theta_2}\left(\tau^{i,j-1}_{op}\right) \\ \|\, \mathrm{MLP}_{\theta_3}\left(\tau^{i,j}_{op}\right) \|\, \mathrm{MLP}_{\theta_4}\left(\tau^{i,j+1}_{op}\right)] \end{array} \right) \tag{29}$$

Embedding mechanism

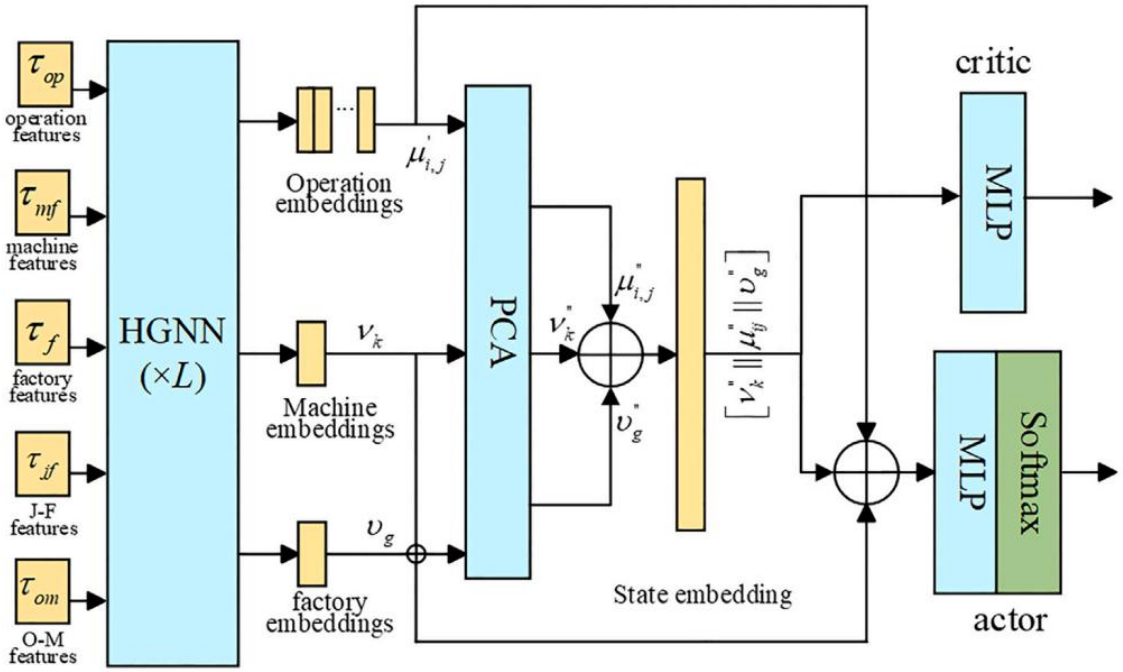


Fig. 1. Structure of the proposed HGNP.

Factory node embedding (GAT)

The GAT method is used for factory node embedding. The detailed factory node embedding can be described as follows.

Step 1. Calculate the attention coefficients on the edge between F_g and τ_{jf}^1 as follows.

$$e_{ig} = \text{LeakyReLu}\left(a^{\mathrm{T}}[W^F \tau_f^g \parallel W^J \tau_{jf}^i]\right) \tag{30}$$

where, $W^F \in R^{d\times 2}$, $W^J \in R^{d\times 1}$, $\tau_f^G \in R^2$, and $a^T \in R^{2d}$.

Step 2. Calculate the normalized attention coefficients as follows.

$$\alpha_{ig} = \sigma\left(Norm\left(e_{ig}\right)\right) \tag{31}$$

Step 3. The factory node embedding v_g for F_g is calculated as follows.

$$v_g = \sigma\left(\sum_i \alpha_{ig} W^J \tau_{jf}^i\right) \tag{32}$$

Embedding mechanism

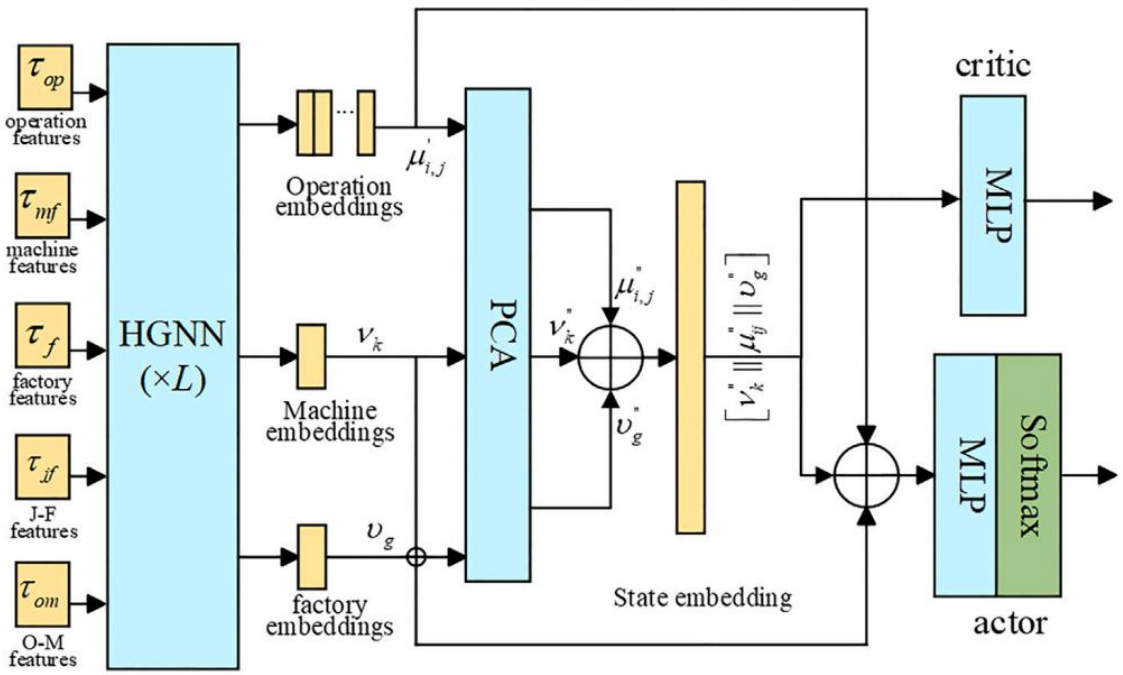


Fig. 1. Structure of the proposed HGNP.

Algorithm 2: PCA-based transform procedure

Input: the three embeddings, i.e., v_k, $\mu'_{i,j}$, and v_g, the embedding length n

. **Output:** the resulting embeddings, v''_k, $\mu''_{i,j}$, and v''_g.

Compute the mean values $\overline{v_k}$, $\overline{\mu_{i,j}}$, and $\overline{v_g}$, for the three embeddings, i.e., v_k, $\mu'_{i,j}$, and v_g, respectively.

Compute U, S, V for v_k, $\mu'_{i,j}$, and v_g as steps 4 to 13.

Generate a random matrix $R_{n\times q}$, $q = min(n, 6)$.

$Q'_m = QR((v_k - \overline{v_k}) \times R_{n\times q})$.

for $i = 1 \rightarrow \ell$ **do**

  $Q''_m = QR((v_k - \overline{v_k})^T Q'_m)$.

  $Q_m = QR((v_k - \overline{v_k}) Q''_m)$.

end

$U_m, S_m, V_m = SVD((v_k - \overline{v_k})^T Q_m)$.

$v'_k = v_k \times V_m$.

$\mu_{i,j} = \mu'_{i,j} \times V_j$.

$v'_g = v_g \times V_g$

Let $v''_k = \text{mean}(v'_k)$, $\mu''_{i,j} = \text{mean}(\mu_{i,j})$, $v''_g = \text{mean}(v'_g)$

Enhanced local search

Algorithm 3: Enhanced local search procedure

Input: a solution s_i generated by the policy network

. **Output:** the improved solution $\overline{s_i}$

Randomly generate P_s solutions, and replace one with s_i.

for $i = 1 \rightarrow \ell_1$ **do**

    Perform partially-matched crossover (PMX), three types of mutations (swap, insertion, inversion) for the population.

    Update the best solution found so far.

end

Record the best solution found so far as $\overline{s_i}$.

Contents

Experiment Results——Evaluation instances

①

Table 4
Problem scales.

Size $n \times m \times h$	θ_j	$\|M_{j,q}\|$	$P_{j,q}$	$ST_{i,i'}$	g
$10 \times 5 \times 2$	$U(4,6)$	$U(1,5)$	$U(1,20)$	$U(1,5)$	2
$20 \times 5 \times 2$	$U(4,6)$	$U(1,5)$	$U(1,20)$	$U(1,5)$	4
$15 \times 10 \times 2$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	3
$20 \times 10 \times 2$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	4
$30 \times 10 \times 2$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	6
$40 \times 10 \times 2$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	8
$10 \times 5 \times 3$	$U(4,6)$	$U(1,5)$	$U(1,20)$	$U(1,5)$	2
$20 \times 5 \times 3$	$U(4,6)$	$U(1,5)$	$U(1,20)$	$U(1,5)$	4
$15 \times 10 \times 3$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	3
$20 \times 10 \times 3$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	4
$30 \times 10 \times 3$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	6
$40 \times 10 \times 3$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	8
$10 \times 5 \times 5$	$U(4,6)$	$U(1,5)$	$U(1,20)$	$U(1,5)$	2
$20 \times 5 \times 5$	$U(4,6)$	$U(1,5)$	$U(1,20)$	$U(1,5)$	4
$15 \times 10 \times 5$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	3
$20 \times 10 \times 5$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	4
$30 \times 10 \times 5$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	6
$40 \times 10 \times 5$	$U(8,12)$	$U(1,10)$	$U(1,20)$	$U(1,5)$	8

②

The second type of instance is used to make comparisons with the MIP, in which a set of small-scale instances with two distributed factories are tested based on the above generation methods. The instances include seven different problem sizes, i.e., (4, 2), (4, 3), (4, 4), (6, 2), (6, 3), (6, 4), and (6, 5), where the two numbers represent the number of jobs and machines, respectively.

③

The third type of instance is extended from two well-known FJSP benchmarks, including the 10 mk instances (mk01-mk10) in Hurink, Jurisch, and Thole (1994); and the three groups of la instances in Behnke and Geiger (2012), Hurink et al. (1994) (rdata, edata and vdata), where each group has 40 instances. The dataset of the distributed factories, setup time, and family information is set based on the above generation methods discussed in the first type of instance.

Experiment Results——Compared algorithms

The compared algorithms include three types. The first type includes heuristics such as FIFO (first-in-first-out), SPT (shortest processing time), LPT (longest processing time). The second type includes the deep learning method HGNN (Song et al., 2023), which is the basic framework extended by the proposed HGNP algorithm. The third type includes meta-heuristics, such as HGA (Tutumlu & Saraç, 2023), and HGA-VNS (Sun et al., 2023). The main reasons to select these compared algorithms are as follows: (1) FIFO, LPT, and SPT are typical heuristics with simple and fast speed, which are generally chosen as compared algorithms with RL or DRL methods; (2) HGA and HGA-VNS are two recently published method for solving different types of FJSPs; and (3) HGNN is a recently published DRL method for solving FJSPs.

The gap value is calculated as follows:

$$gap = (C_{\max} - C_B)/C_B \times 100\% \tag{36}$$

Experiment Results

Table 6
Comparisons with CPLEX.

Inst	Scale	Best	Makespan		Gap	
			CPLEX	HGNP	CPLEX	HGNP
1	4 × 2 × 2	22	22	22	0.00+	0.00+
2	4 × 2 × 2	22	22	22	0.00+	0.00+
3	4 × 2 × 2	28	28	28	0.00+	0.00+
4	4 × 3 × 2	17	18	17	5.88	0.00+
5	4 × 3 × 2	29	54	29	86.21	0.00+
6	4 × 3 × 2	18	18	18	0.00+	0.00+
7	4 × 4 × 2	36	37	36	2.78	0.00+
8	4 × 4 × 2	22	22	23	0.00+	4.55
9	4 × 4 × 2	32	32	34	0.00+	6.25
10	6 × 2 × 2	30	30	33	0.00+	10.00
11	6 × 2 × 2	25	25	25	0.00+	0.00+
12	6 × 2 × 2	30	30	32	0.00+	6.67
13	6 × 3 × 2	24	24	30	0.00+	25.00
14	6 × 3 × 2	29	29	34	0.00+	17.24
15	6 × 3 × 2	35	35	38	0.00+	8.57
16	6 × 4 × 2	43	44	43	2.33	0.00+
17	6 × 4 × 2	33	33	37	0.00+	12.12
18	6 × 4 × 2	42	50	42	19.05	0.00+
19	6 × 5 × 2	37	52	37	40.54	0.00+
20	6 × 5 × 2	22	–	49	–	0.00+
Avg		29.16	31.84	30.53	7.87	4.52+

- +means the better value.

Experiment Results——Generalization performance on large-sized instances

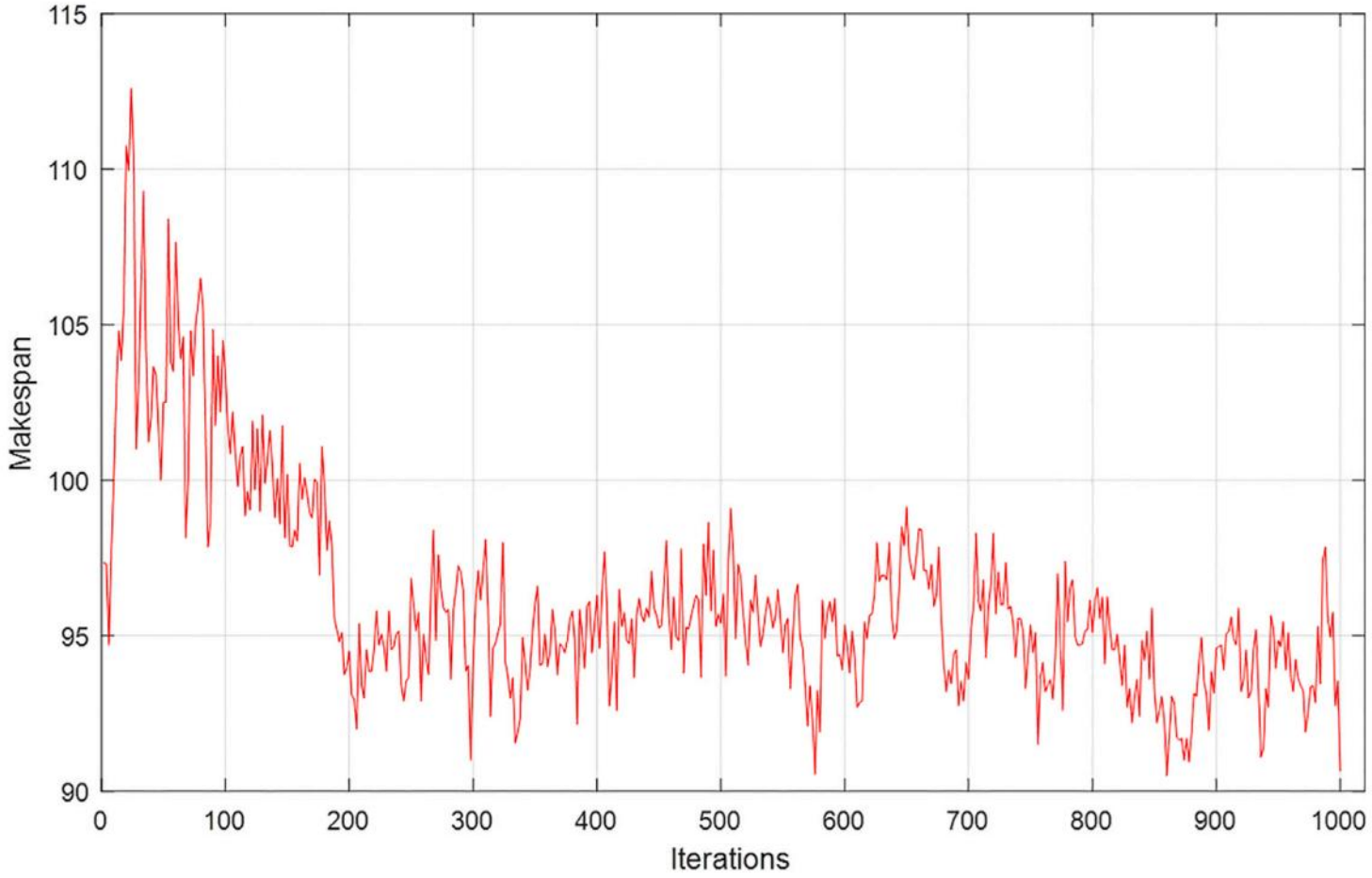


Fig. 3. Training curve on $10 \times 5 \times 2$ instances.

Experiment Results——Generalization performance on large-sized instances

Table 7
Comparisons of the generalization performance.

Model	Problem Scale	Best	Makespan		Gap	
			HGNN	HGNP	HGNN	HGNP
$M_{10\times5\times2}$	50 × 10 × 2	485.35	494.87	485.35	1.96	0.00+
	60 × 10 × 2	561.26	572.70	561.26	2.04	0.00+
	70 × 10 × 2	643.64	647.64	643.64	0.62	0.00+
	70 × 20 × 2	780.06	797.29	780.06	2.21	0.00+
	80 × 10 × 2	723.33	729.59	723.33	0.87	0.00+
	80 × 20 × 2	871.69	884.85	871.69	1.51	0.00+
	80 × 30 × 2	1054.20	1112.56	1054.20	5.54	0.00+
$M_{10\times5\times3}$	50 × 10 × 3	387.52	396.83	387.52	2.40	0.00+
	60 × 10 × 3	415.28	433.12	415.28	4.30	0.00+
	70 × 10 × 3	489.33	521.12	489.33	6.50	0.00+
	70 × 20 × 3	621.15	621.15	622.83	0.00+	0.27
	80 × 10 × 3	675.67	699.34	675.67	3.50	0.00+
	80 × 20 × 3	733.48	745.61	733.48	1.65	0.00+
	80 × 30 × 3	887.15	903.14	887.15	1.80	0.00+
$M_{10\times5\times5}$	50 × 10 × 5	289.19	321.18	289.19	11.06	0.00+
	60 × 10 × 5	356.47	377.49	356.47	5.90	0.00+
	70 × 10 × 5	392.91	411.69	392.91	4.78	0.00+
	70 × 20 × 5	479.87	501.65	479.87	4.54	0.00+
	80 × 10 × 5	598.79	598.79	601.12	0.00+	0.39
	80 × 20 × 5	633.73	633.73	651.34	0.00+	2.78
	80 × 30 × 5	769.12	785.73	769.12	2.16	0.00+
Avg		611.87	628.10	612.90	3.02	0.16+

- +means the better value.

Experiment Results——Efficiency of the enhanced local search heuristic

Table 8
Comparisons of the enhanced local search heuristic.

Size	Best	Makespan		Gap	
		HGNP-NL	HGNP	HGNP-NL	HGNP
15 × 10 × 2	164.97	171.16	164.97	3.75	0.00+
20 × 5 × 2	148.95	154.21	148.95	3.53	0.00+
20 × 10 × 2	183.77	185.05	183.77	0.70	0.00+
30 × 10 × 2	242.87	245.57	242.87	1.11	0.00+
40 × 10 × 2	298.63	299.77	298.63	0.38	0.00+
15 × 10 × 3	119.13	125.65	119.13	5.47	0.00+
20 × 5 × 3	132.67	136.88	132.67	3.17	0.00+
20 × 10 × 3	147.19	149.27	147.19	1.41	0.00+
30 × 10 × 3	185.88	187.8	185.88	1.03	0.00+
40 × 10 × 3	224.22	225.64	224.22	0.63	0.00+
15 × 10 × 5	105.22	108.34	105.22	2.97	0.00+
20 × 5 × 5	109.12	115.23	109.12	5.60	0.00+
20 × 10 × 5	129.29	134.01	129.29	3.65	0.00+
30 × 10 × 5	153.11	155.92	153.11	1.84	0.00+
40 × 10 × 5	195.59	196.77	195.59	0.60	0.00+
Avg	169.37	170.15	169.37	2.40	0.00+

- +means the better value.

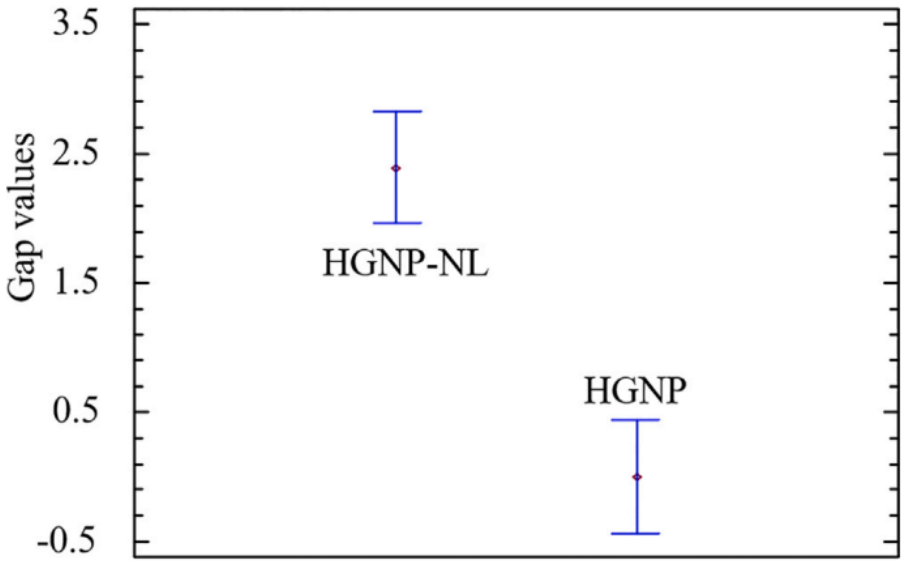


Fig. 4. ANOVA comparisons with HGNP and HGNP-NL.

Experiment Results——Efficiency of the PCA-based transform mechanism

Table 9
Comparisons of the PCA-based transform mechanism.

Size	Best	Makespan		Gap	
		HGNN	HGNP-NL	HGNN	HGNP-NL
15 × 10 × 2	171.16	171.20	171.16	0.02	+0.00
20 × 5 × 2	154.21	176.36	154.21	14.36	+0.00
20 × 10 × 2	185.05	189.54	185.05	2.43	+0.00
30 × 10 × 2	245.57	268.49	245.57	9.33	+0.00
40 × 10 × 2	299.77	341.90	299.77	14.05	+0.00
15 × 10 × 3	123.1	123.10	125.65	+0.00	2.07
20 × 5 × 3	136.88	139.93	136.88	2.23	+0.00
20 × 10 × 3	149.27	157.18	149.27	5.30	+0.00
30 × 10 × 3	187.8	197.22	187.8	5.02	+0.00
40 × 10 × 3	225.64	229.16	225.64	1.56	+0.00
15 × 10 × 5	107.25	107.25	108.34	+0.00	1.02
20 × 5 × 5	115.23	117.18	115.23	1.69	+0.00
20 × 10 × 5	134.01	138.39	134.01	3.27	+0.00
30 × 10 × 5	155.92	158.93	155.92	1.93	+0.00
40 × 10 × 5	196.77	200.16	196.77	1.72	+0.00
Avg	172.51	181.07	170.15	4.19	+0.21

- +means the better value.

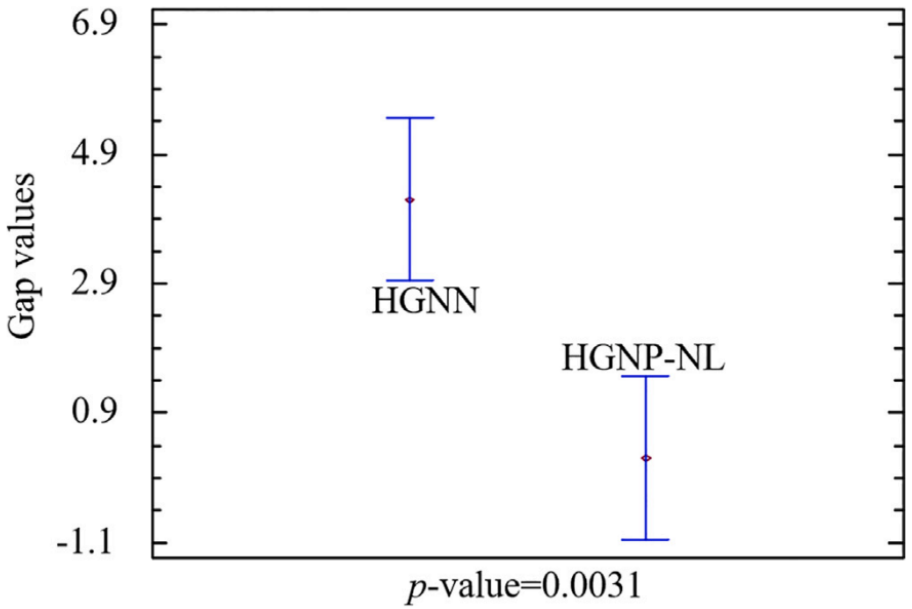


Fig. 5. ANOVA comparisons for HGNN and HGNP-NL.

Experiment Results—— Comparisons with heuristics and meta-heuristics

Table 10
Comparisons with heuristics and meta-heuristics for 2-factory DFJSP (Gap values).

Problem	Size					Avg
	15 × 10 × 2	20 × 5 × 2	20 × 10 × 2	30 × 10 × 2	40 × 10 × 2	
HGA	0.93	+0.00	12.65	9.81	11.63	7.00
HGA-VNS	2.51	1.66	13.53	10.61	11.85	8.03
RANDOM	63.17	54.48	87.56	87.53	90.76	76.70
LPT	14.07	12.00	24.80	17.65	17.82	17.27
SPT	16.53	14.49	27.60	20.36	19.18	19.63
FIFO	16.57	14.90	25.46	19.16	18.92	19.00
HGNP	+0.00	2.66	+0.00	+0.00	+0.00	+0.53

- +means the better value.

Table 11
Comparisons with heuristics and meta-heuristics for 3-factory DFJSP (Gap values).

Problem	Size					Avg
	15 × 10 × 3	20 × 5 × 3	20 × 10 × 3	30 × 10 × 3	40 × 10 × 3	
HGA	0.87	0.23	5.81	10.13	13.45	6.10
HGA-VNS	+0.00	0.13	4.92	8.12	10.42	4.72
RANDOM	76.23	69.34	90.43	95.76	100.34	86.42
LPT	16.71	15.87	30.83	25.3	18.88	21.52
SPT	18.48	16.62	29.92	23.5	20.18	21.74
FIFO	19.66	17.91	24.48	20.1	22.32	20.89
HGNP	0.53	+0.00	+0.00	+0.00	+0.00	+0.11

- +means the better value.

Table 12
Comparisons with heuristics and meta-heuristics for 5-factory DFJSP (Gap values).

Problem	Size					Avg
	15 × 10 × 5	20 × 5 × 5	20 × 10 × 5	30 × 10 × 5	40 × 10 × 5	
HGA	0.52	1.46	5.87	4.98	8.72	4.31
HGA-VNS	+0.00	+0.00	2.91	4.35	5.09	2.47
RANDOM	90.12	78.85	100.19	98.77	120.43	97.67
LPT	29.14	20.33	31.17	28.19	46.01	30.97
SPT	19.93	20.68	35.73	33.89	48.18	31.68
FIFO	19.45	28.81	33.17	35.57	29.76	29.35
HGNP	0.15	0.09	+0.00	+0.00	+0.00	+0.05

- +means the better value.

Experiment Results—— Comparisons of run time

Table 13

Comparisons of total consumed computational times (second).

Problem	Size				
	15 × 10	20 × 5	20 × 10	30 × 10	40 × 10
HGA	4.2E3	2.4E3	5.2E3	6.4 E3	8.5E3
HGA-VNS	5.9E3	2.9 E3	6.3E3	9.8E3	1.5E4
HGNP-NL	3.7E2+	2.5E2+	4.9E2+	8.2E2+	1.7E3+
HGNP	5.8E2	4.3E2	1.5E3	2.3E3	3.2E3

- +means the better value.

Experiment Results—— Comparisons of public instances

Table 14
Comparisons of the extended public instances.

Algorithm	Mk			rdata			edata			vdata		
	C_{max}	Gap	Time(s)	C_{max}	Gap	Time(s)	C_{max}	Gap	Time(s)	C_{max}	Gap	Time(s)
HGA	149.3	5.7%	35.1	721.3	1.1%	29.1	709.1	3.4%	27.6	695.8	8.3%	34.5
HGA-VNS	153.2	8.5%	40.2	713.2	+0.0%	38.9	713.4	4.1%	38.4	694.3	8.1%	42.3
$HGNP_1$	146.1	3.5%	1.8	720.7	1.1%	0.8	746.2	8.8%	1.7	683.7	6.4%	1.8
$HGNP_2$	145.6	3.1%	2.1	718.9	0.8%	1.2	701.9	2.4%	1.9	657.0	2.3%	1.6
$HGNP_3$	150.7	6.7%	1.9	713.7	0.1%	1.1	719.2	4.9%	1.8	656.2	2.1%	1.9
$HGNP_4$	141.2	+0.0%	1.9	717.9	0.7%	1.0	693.4	1.1%	1.9	649.7	1.1%	1.8
$HGNP_5$	145.8	3.3%	1.6	719.5	0.9%	1.0	697.0	1.7%	1.8	659.1	2.6%	1.7
$HGNP_6$	143.2	1.4%	1.8	717.3	0.6%	1.3	685.6	+0.0%	1.6	642.5	+0.0%	1.5

↓ means the better value

Experiment Results—— Comparisons of public instances

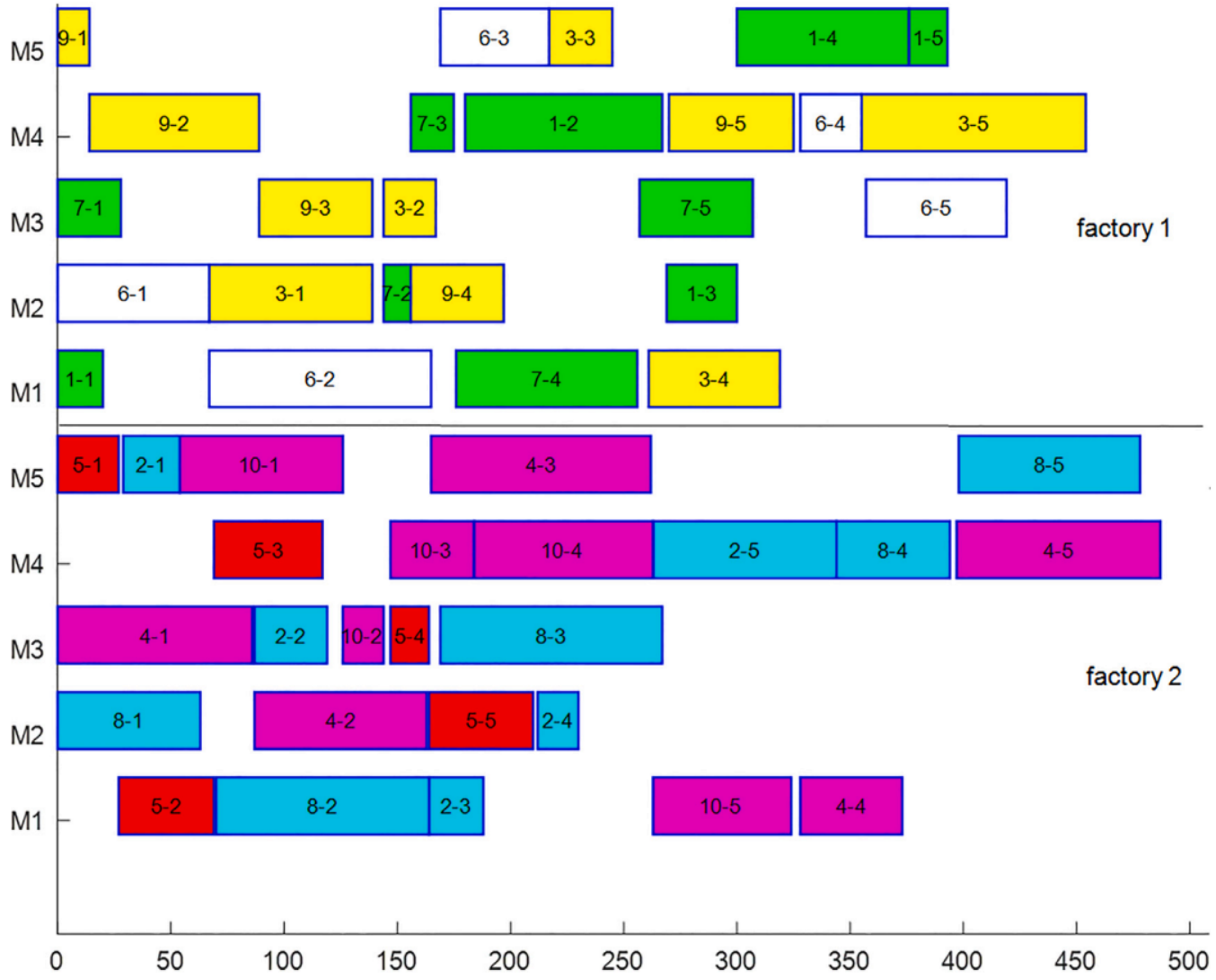


Fig. 6. Gantt chart of one solution for "e_la11" with two factories.

Contents

Future Work

(1) consider more realistic constraints in the DFJSP, such as fuzzy processing time, no-wait, and limited resource constraints;

(2) combine efficient components into the heterogeneous graph neural network, and therefore, the embeddings of different raw features can be fused efficiently;

(3) consider more useful objectives, such as energy consumption, working efficiency, and economic costs;

(4) apply the proposed policy network to solve realistic optimization problems, such as steelmaking casting problems, and prefabricated scheduling problems;

(5) consider generalization simultaneously in the number of distributed factories, jobs and machines.

Paper 3

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

Distributed heterogeneous flexible job-shop scheduling problem considering automated guided vehicle transportation via improved deep Q network

Minghai Yuan * , Songwei Lu , Liang Zheng , Qi Yu , Fengque Pei , Wenbin Gu

Publish Journal: Swarm and Evolutionary Computation

Published Time: March 2, 2025

IF: 8.5

Contents

Contents

Problem Description

This study proposes an improved deep Q network (DQN) real-time scheduling method aimed at minimizing makespan. A mixed integer linear programming model (MILP) of DHFJSP-AGV is developed and transformed into a Markov decision process (MDP). Eight general state features are extracted and normalized to represent the state space, while appropriate combination dispatching rules are selected as the action space. The state features of each scheduling point are input to the DQN, determining the factory, job, machine, and AGV for each process. Additionally, double DQN and an improved ε-greedy exploration are used to enhance the DQN.

Contents

Description of DHFJSP-AGV

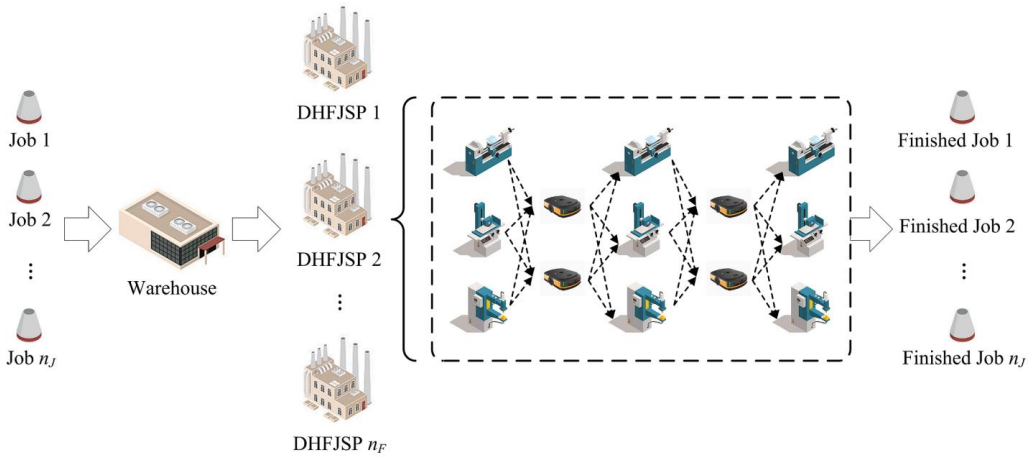


Fig. 1. Schematic diagram of DHFJSP-AGV.

Table 1
Processing time of four jobs in two heterogeneous factories.

Job	Operation	F_1			F_2			AGV
		M_{11}	M_{12}	M_{13}	M_{21}	M_{22}	M_{23}	
J_1	O_{11}	–	18	23	–	17	20	A_{11}
	O_{12}	27	–	30	24	18	–	A_{21}
J_2	O_{21}	–	27	21	–	33	25	
	O_{22}	19	30	–	21	25	–	
J_3	O_{31}	33	–	29	21	–	18	
	O_{32}	–	30	25	27	19	–	
	O_{33}	24	–	24	–	24	19	
J_4	O_{41}	19	–	30	25	17	21	
	O_{42}	18	25	–	19	–	24	

Table 2
Transportation time between different machines in DHFJSP-AGV.

	LU	M_1	M_2	M_3
LU	0	6	5	6
M_1	3	0	2	7
M_2	6	10	0	4
M_3	7	4	8	0

As shown in Fig. 1, DHFJSP-AGV can be described as: n_J jobs are processed in n_F heterogeneous factories. Each factory has n_M machines and n_A AGVs. The number of machines and AGVs in every factory is the same, but the processing capacity of machines is different. After the job is assigned to the factory, it cannot be transferred to other factories for processing. Each job contains n_{J_i} operations that should be processed according to the predetermined sequence. Each operation can be processed on a set of candidate machines $\mathscr{M}_{ij}$. The job is transported to different machines for processing through AGV. In the DHFJSP-AGV, it is necessary to reasonably select factories, jobs, machines and AGVs to achieve the scheduling goal of minimizing the makespan.

Description of DHFJSP-AGV

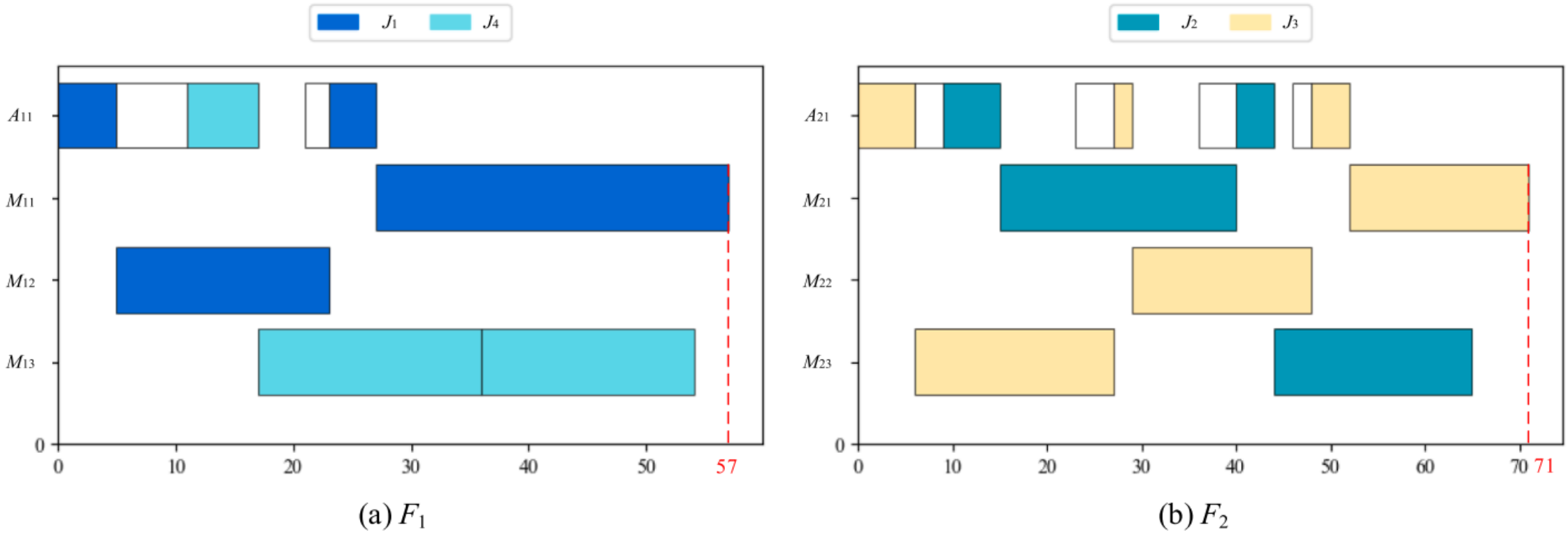


Fig. 2. A feasible scheduling Gantt chart of DHFJSP-AGV.

MDP——state space and reward

Factory

- The average completion rate of jobs
- The standard deviation of the job completion rate

Job

- The average completion rate of the operations
- The standard deviation of the operation completion rate

Machine

- The average utilization rate of machines
- The standard deviation of the machine utilization rate

AGV

- The average utilization rate of AGVs
- The standard deviation of the AGV utilization rate

Reward

$$r_t = CTJ_{\max} - \max_{i=1,2,\cdots,n_J} (CTJ_i)$$

MDP——action space

Table 5
The set of dispatching rules.

Type	Name	Description
Factory dispatching rule	$MinCT_F$	Select the factory with the shortest completion time
	$MaxCT_F$	Select the factory with the longest completion time
	$MinNJ_F$	Select the factory with the least allocated jobs
	$MaxNJ_F$	Select the factory with the most allocated jobs
Job dispatching rule	SPT_J	Select the job with the shortest processing time
	LPT_J	Select the job with the longest processing time
	$SPTR_J$	Select the job with the shortest remaining processing time
	$LPTR_J$	Select the job with the longest remaining processing time
	$SPTSO_J$	Select the job with the shortest processing time of subsequent operation
	$LPTSO_J$	Select the job with longest processing time of subsequent operation
	$LRUO_J$	Select the job with the least remaining unprocessed operations
	$MRUO_J$	Select the job with the most remaining unprocessed operations
Machine dispatching rule	SPT_M	Select the machine with the shortest processing time
	LPT_M	Select the machine with the longest processing time
	LU_M	Select the machine with the lowest utilization rate
	HU_M	Select the machine with the highest utilization rate
	$MinCT_M$	Select the machine with the minimum completion time
	$MaxCT_M$	Select the machine with the maximum completion time
AGV dispatching rule	STT_A	Select the AGV with the shortest transportation time
	LTT_A	Select the AGV with the longest transportation time
	LU_A	Select the AGV with the lowest utilization rate
	HU_A	Select the AGV with the highest utilization rate
	$MinCT_A$	Select the AGV with the minimum completion time
	$MaxCT_A$	Select the AGV with the maximum completion time

Scheduling framework based on improved DQN

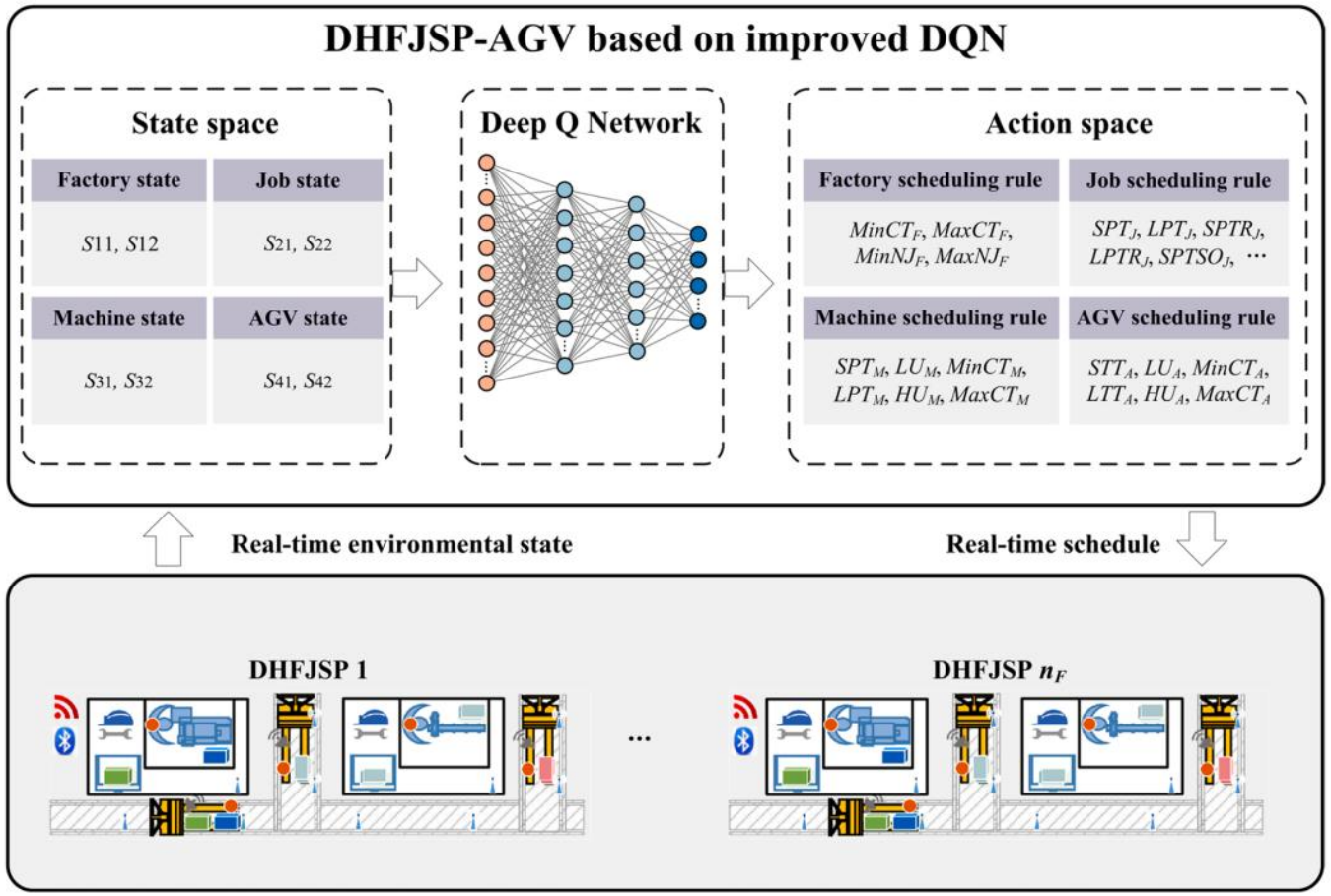


Fig. 3. Scheduling framework of DHFJSP-AGV based on improved DQN.

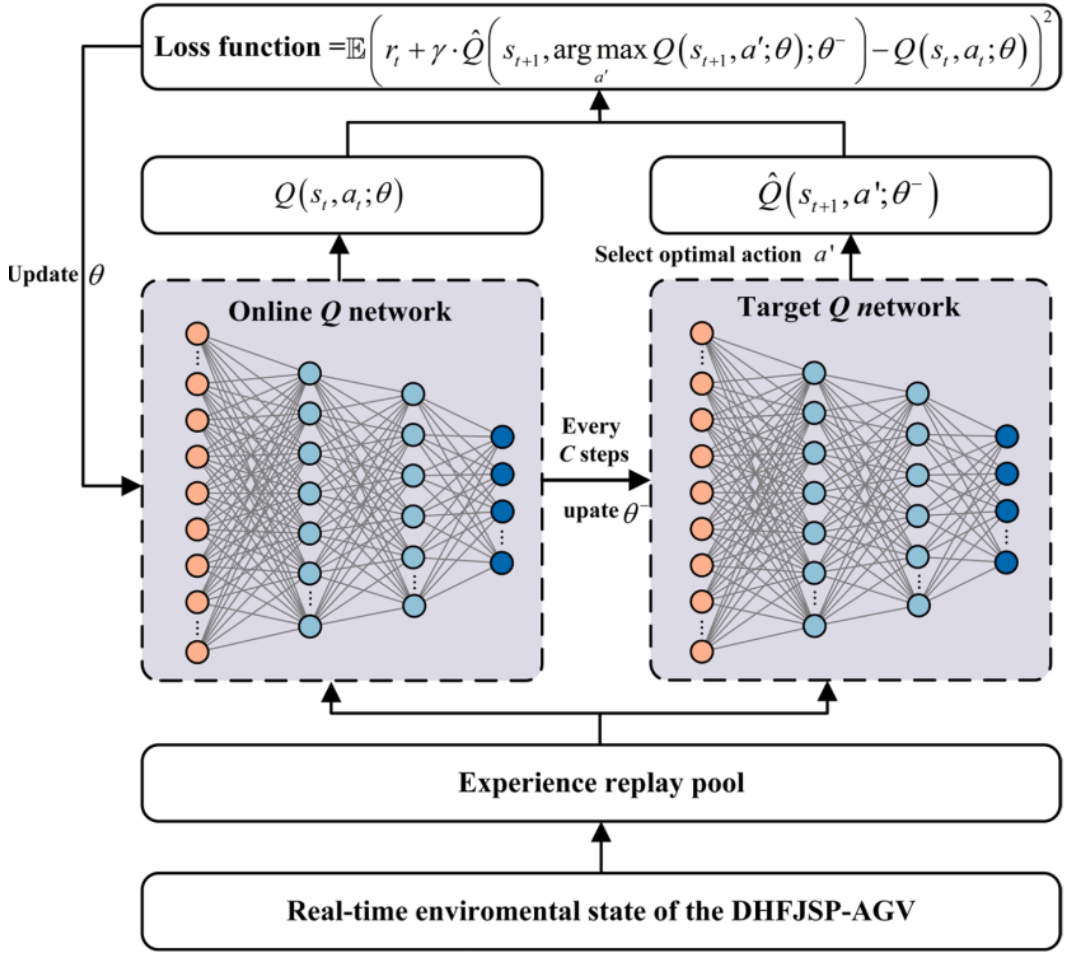


Fig. 4. Structure of the improved DQN.

Contents

Experiment Results--Test instances

Since DHFJSP-AGV is not studied before, and no public dataset is available. Therefore, this study generates 42 test instances based on reference [16], with the dataset accessible at: https://github.com/Hinachanyasashii/DHFJSP_AGV_DQN.git. In the generated instances, the number of factories ranges from $n_F \in \{2,3,4,5,6,7,8\}$, and the number of jobs ranges from $n_J \in \{10,20,30,40,50,80,100,120,150,200,250,300\}$. Each factory contains either 5 or 10 machines, reflecting different factory sizes. Each job consists of five operations, and each operation can be processed by one or more compatible machines. The processing time for each operation falls within the range $pt_{ijfk} \in [5,20]$, with varying times for the same operation across different factories. Jobs are transported from the starting location LU to different machines for processing, with the transportation time $tt_{kk'}$ generated by a random function. Since the machine layout is identical across factories, a shared transportation time matrix is used, as shown in Eq. (31).

$$TT = \begin{bmatrix} 0 & 6 & 8 & 6 & 8 & 10 & 12 & 10 & 12 & 8 \\ 8 & 0 & 2 & 8 & 2 & 4 & 6 & 4 & 6 & 8 \\ 6 & 10 & 0 & 10 & 8 & 2 & 4 & 6 & 4 & 10 \\ 12 & 4 & 6 & 0 & 6 & 8 & 10 & 8 & 10 & 6 \\ 10 & 2 & 4 & 6 & 0 & 6 & 8 & 2 & 8 & 2 \\ 8 & 8 & 2 & 8 & 6 & 0 & 6 & 4 & 2 & 8 \\ 6 & 10 & 8 & 10 & 8 & 6 & 0 & 6 & 4 & 4 \\ 12 & 4 & 6 & 4 & 2 & 8 & 10 & 0 & 10 & 4 \\ 10 & 6 & 4 & 6 & 4 & 2 & 8 & 2 & 0 & 10 \\ 2 & 12 & 4 & 10 & 12 & 4 & 10 & 2 & 4 & 0 \end{bmatrix}$$

Experiment Results--The training process of improved DQN

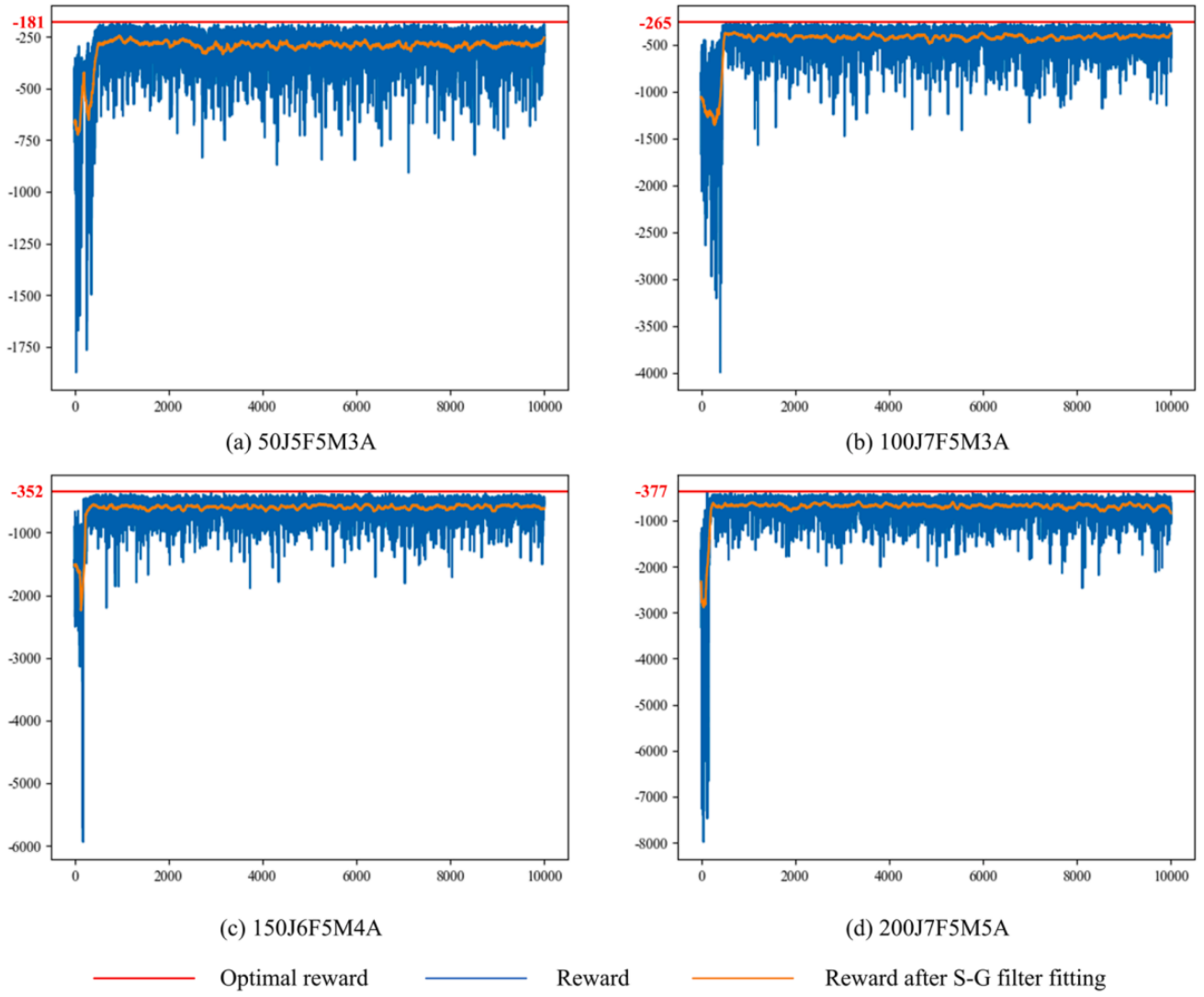


Fig. 5. Reward convergence curves of 50J5F5M3A, 100J7F5M3A, 150J6F5M4A and 200J7F5M5A.

Experiment Results--Comparisons with the well-known combination dispatching rules

1) CDR1: $MinNJ_F + SPT_J + MinCT_M + MinCT_A$, Select the factory with the least allocated jobs, the job with the shortest processing time, the machine with the minimum completion time, and the AGV with the minimum completion time.
2) CDR2: $MinNJ_F + SPT_J + SPT_M + MinCT_A$, Select the factory with the least allocated jobs, the job with the shortest processing time, the machine with the shortest processing time, and the AGV with the minimum completion time.
3) CDR3: $MinNJ_F + SPTR_J + MinCT_M + MinCT_A$, Select the factory with the allocated jobs, the job with the shortest remaining processing time, the machine with the minimum completion time, and the AGV with the minimum completion time.
4) CDR4: $MinNJ_F + SPTR_J + SPT_M + MinCT_A$, Select the factory with the least allocated jobs, the job with the shortest remaining processing time, the machine with the shortest processing time, and the AGV with the minimum completion time.
5) CDR5: $MinNJ_F + MRUO_J + MinCT_M + MinCT_A$, Select the factory with the least allocated jobs, the job with the most remaining unprocessed operations, the machine with the shortest completion time, and the AGV with the minimum completion time.
6) CDR6: $MinNJ_F + MRUO_J + SPT_M + MinCT_A$, Select the factory with the least allocated jobs, the job with the most remaining unprocessed operations, the machine with the shortest processing time, and the AGV with the minimum completion time.

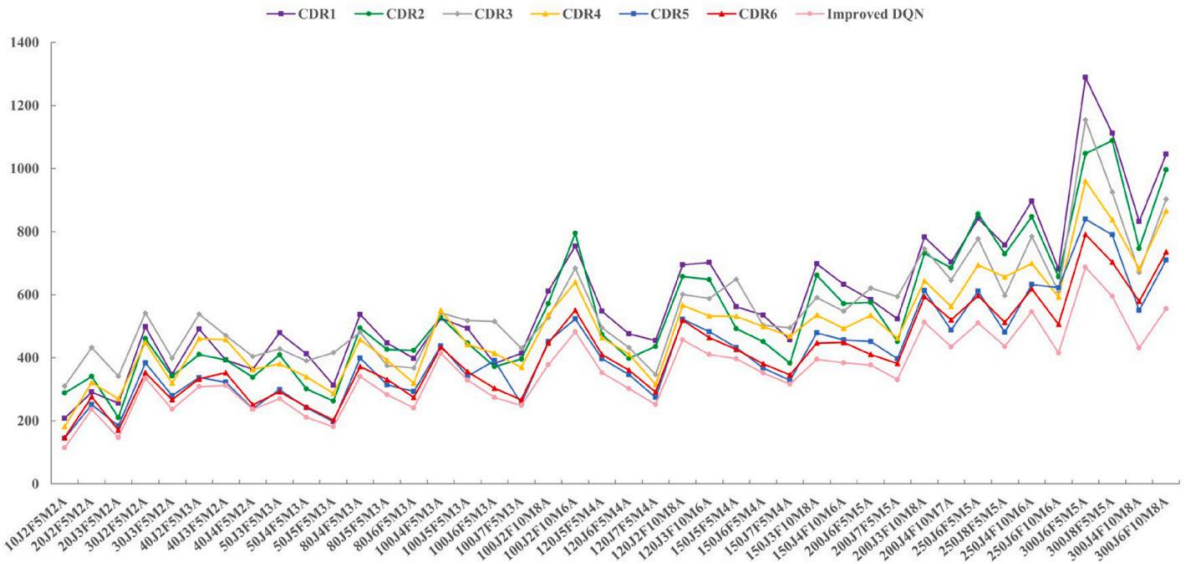
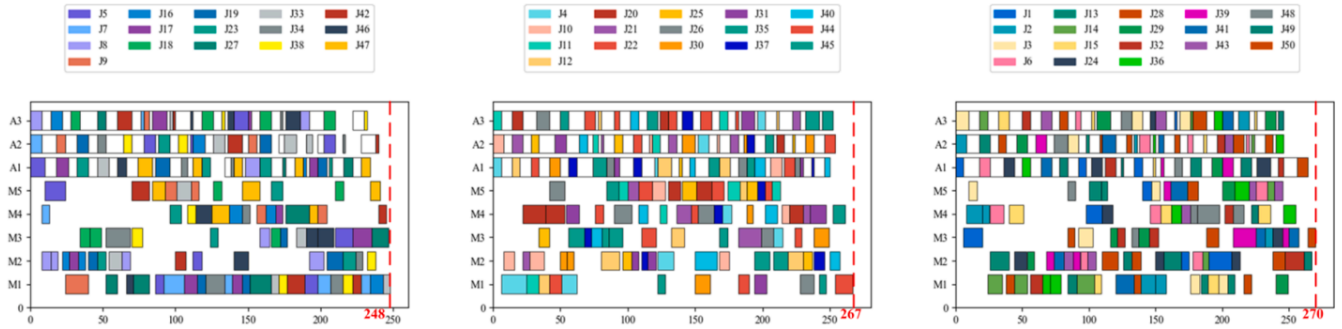


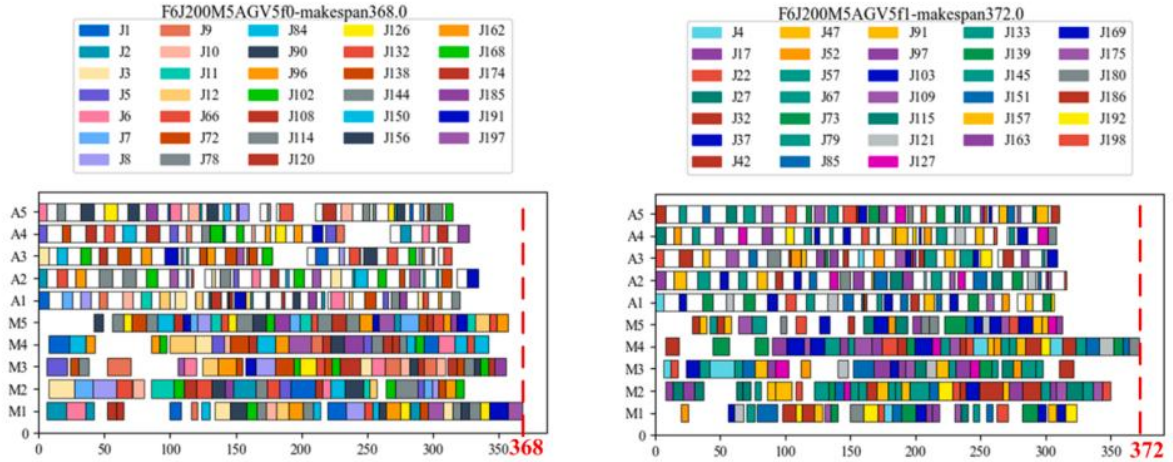
Fig. 6. Comparison of improved DQN and combination dispatching rules.

Experiment Results--Comparisons with the well-known combination dispatching rules

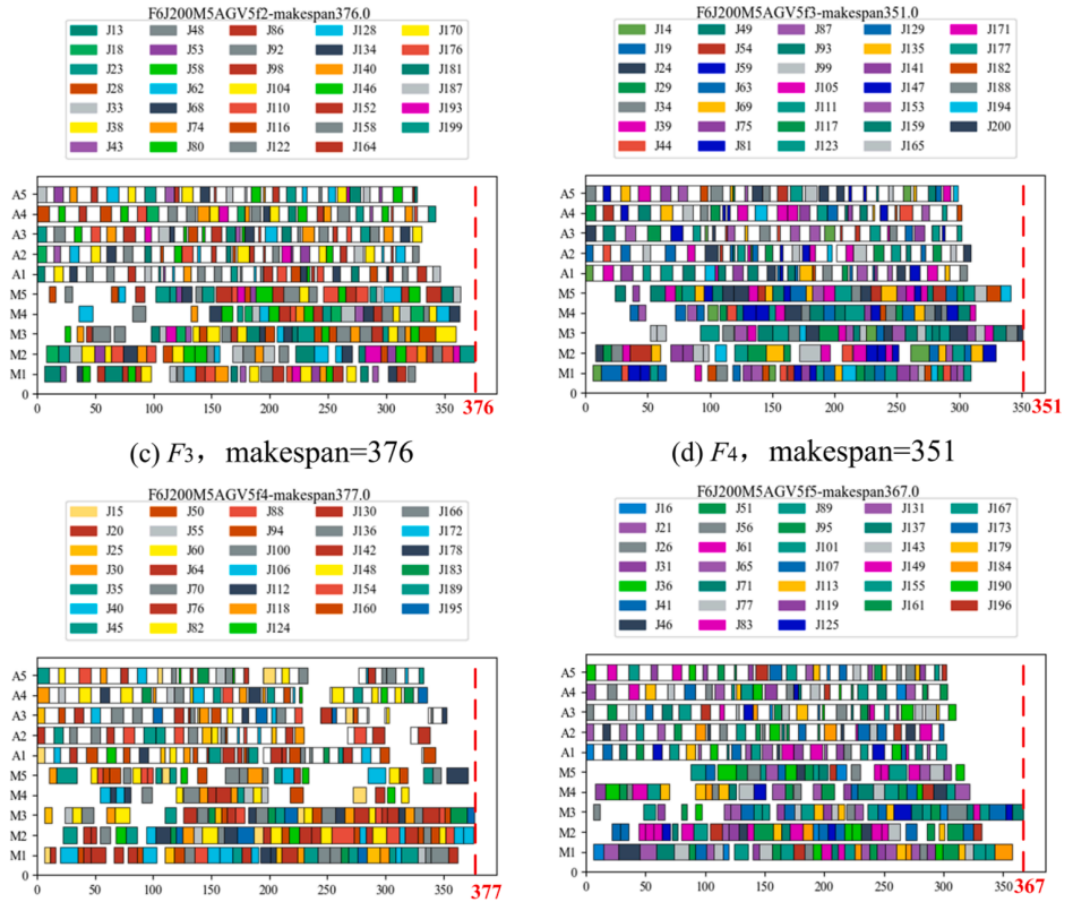


(a) F_1， makespan=248　(b) F_2， makespan=267　(c) F_3， makespan=270

Fig. 7. Scheduling Gantt chart of the 50J 3F5M3A.



(a) F_1， makespan=368　(b) F_2， makespan=372



(c) F_3， makespan=376　(d) F_4， makespan=351

(e) F_5， makespan=377　(f) F_6， makespan=367

Fig. 8. Scheduling Gantt chart of the 200J6F5M5A.

Experiment Results--Comparisons with the well-known combination dispatching rules

Table 8
Mean value of makespan and training time by basic DQN, double DQN and improved DQN.

Instance	Basic DQN		Double DQN		Improved DQN	
	Objective/Gap	Time (s)	Objective/Gap	Time (s)	Objective/Gap	Time (s)
10J2F5M2A	137/20.18 %	0.07	119*/4.39 %	0.07	**114/0.00 %**	0.08
20J2F5M2A	258/13.16 %	0.13	242/6.14 %	0.14	**228/0.00 %**	0.17
20J3F5M2A	192/30.61 %	0.14	151*/2.72 %	0.15	**147/0.00 %**	0.18
30J2F5M2A	376/13.25 %	0.18	376/13.25 %	0.20	**332/0.00 %**	0.26
30J3F5M2A	276/18.97 %	0.19	244*/5.17 %	0.22	**232/0.00 %**	0.27
40J2F5M3A	358/14.74 %	0.31	333/6.73 %	0.33	**312/0.00 %**	0.37
40J3F5M2A	391/25.32 %	0.31	339/8.65 %	0.34	**312/0.00 %**	0.38
40J4F5M2A	300/27.12 %	0.32	247*/4.66 %	0.34	**236/0.00 %**	0.38
50J 3F5M3A	399/47.78 %	0.40	291/7.78 %	0.43	**270/0.00 %**	0.48
50J4F5M3A	304/42.06 %	0.40	260/21.50 %	0.45	**214/0.00 %**	0.49
50J5F5M3A	262/53.21 %	0.39	204/19.30 %	0.44	**171/0.00 %**	0.50
80J4F5M3A	470/37.83 %	0.52	395/15.84 %	0.55	**341/0.00 %**	0.61
80J5F5M3A	388/37.10 %	0.52	351/24.03 %	0.55	**283/0.00 %**	0.61
80J6F5M3A	355/47.30 %	0.51	302/25.31 %	0.54	**241/0.00 %**	0.60
100J4F5M3A	670/61.45 %	0.71	553/33.25 %	0.83	**415/0.00 %**	1.09
100J5F5M3A	587/73.15 %	0.74	487/43.66 %	0.85	**339/0.00 %**	1.08
100J6F5M3A	427/50.35 %	0.76	396/39.44 %	0.88	**284/0.00 %**	1.10
100J7F5M3A	339/33.46 %	0.94	303/19.29 %	1.12	**254/0.00 %**	1.15
100J2F10M8A	588/55.56 %	1.10	512/35.45 %	1.15	**378/0.00 %**	1.23
100J3F10M6A	509/52.40 %	1.11	470/40.72 %	1.15	**334/0.00 %**	1.26
120J5F5M4A	525/48.73 %	1.13	472/33.71 %	1.25	**353/0.00 %**	1.47
120J6F5M4A	442/46.36 %	1.13	385/27.48 %	1.26	**302/0.00 %**	1.47
120J7F5M4A	423/67.86 %	1.12	339/34.52 %	1.25	**252/0.00 %**	1.49
120J2F10M8A	690/50.98 %	1.22	623/36.32 %	1.43	**457/0.00 %**	1.79
120J3F10M6A	648/58.05 %	1.29	591/44.15 %	1.50	**410/0.00 %**	1.83
150J5F5M4A	623/56.93 %	1.25	564/42.07 %	1.47	**397/0.00 %**	1.88
150J6F5M4A	527/54.09 %	1.35	487/42.40 %	1.53	**342/0.00 %**	1.91
150J7F5M4A	501/63.19 %	1.64	441/43.65 %	1.79	**307/0.00 %**	1.96
150J3F10M8A	627/53.13 %	1.98	571/44.56 %	2.11	**395/0.00 %**	2.28
150J4F10M6A	588/51.44 %	2.04	554/44.27 %	2.18	**384/0.00 %**	2.39
200J6F5M5A	634/65.53 %	2.36	580/51.44 %	2.44	**383/0.00 %**	2.73
200J7F5M5A	603/72.28 %	2.40	574/64.00 %	2.60	**350/0.00 %**	2.86
200J3F10M8A	887/73.24 %	3.13	796/55.47 %	3.31	**512/0.00 %**	3.56
200J4F10M7A	722/65.98 %	3.21	651/49.66 %	3.37	**435/0.00 %**	3.60
250J6F5M5A	832/63.14 %	3.73	753/47.65 %	3.75	**510/0.00 %**	3.84
250J8F5M5A	801/83.72 %	3.70	711/63.07 %	3.75	**436/0.00 %**	3.82
250J4F10M6A	975/78.57 %	3.96	798/46.15 %	4.09	**546/0.00 %**	4.22
250J6F10M6A	771/85.78 %	3.92	665/60.24 %	4.04	**415/0.00 %**	4.13
300J6F5M5A	1259/83.26 %	6.27	1102/60.41 %	6.50	**687/0.00 %**	6.98
300J8F5M5A	1084/82.35 %	5.84	1043/75.29 %	6.23	**595/0.00 %**	6.61
300J4F10M8A	836/93.96 %	6.82	725/68.21 %	7.01	**431/0.00 %**	7.28
300J6F10M8A	987/77.52 %	6.53	844/62.59 %	6.79	**556/0.00 %**	7.04

Experiment Results--Comparisons with the well-known combination dispatching rules

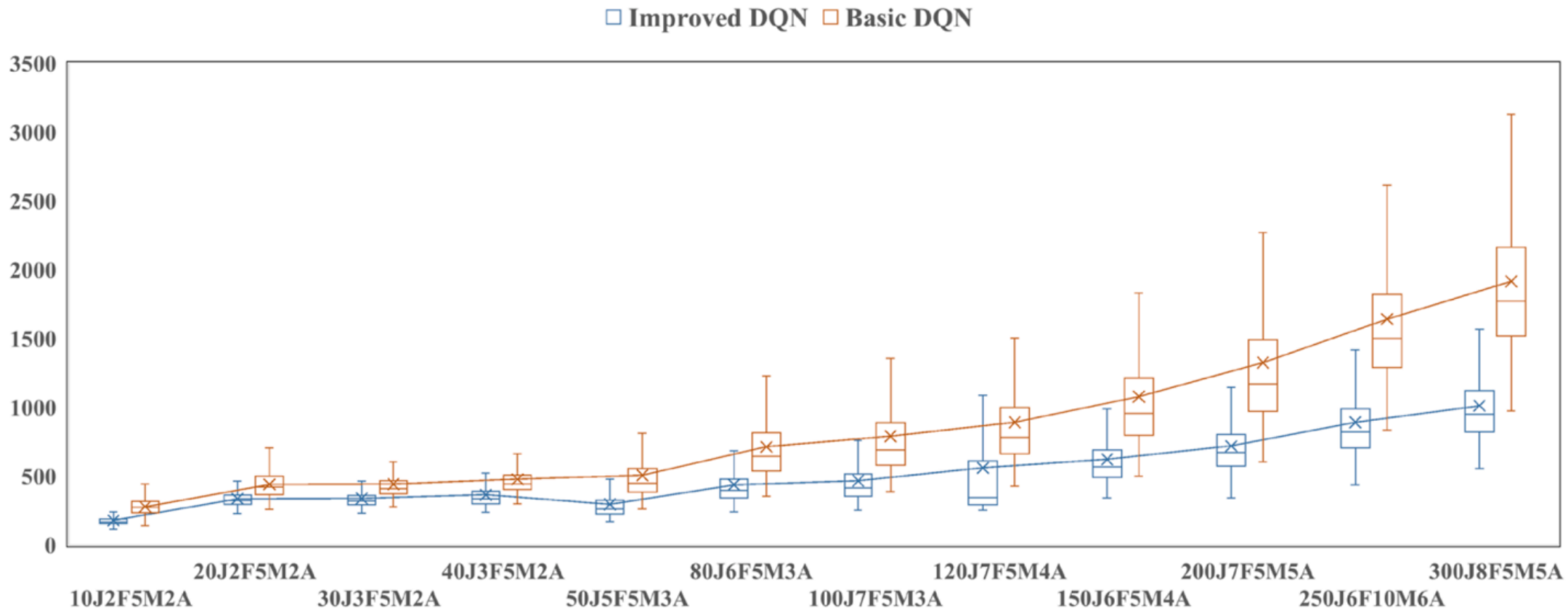


Fig. 9. Box plots of makespan for different instances.

Experiment Results--Example verification

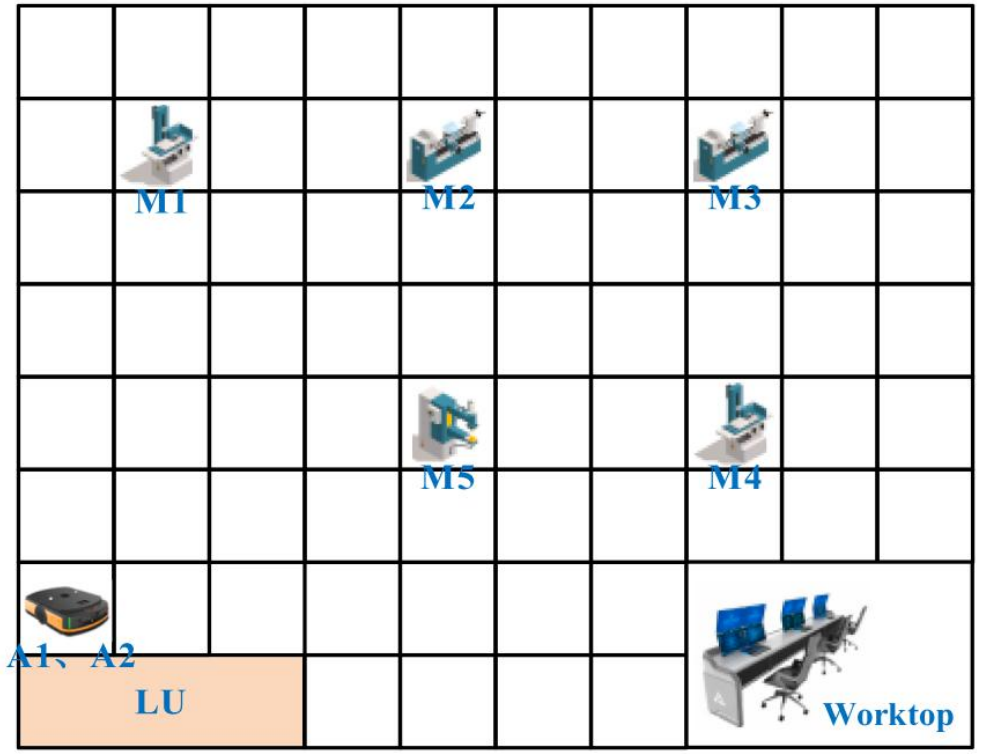


Fig. 10. Grid map of the workshop.

Table 9
The coordinate position of each device.

Device	Coordinate	Device	Coordinate
A1	(0, 2)	M3	(14, 12)
A2	(0, 2)	M4	(14, 6)
M1	(2, 12)	M5	(8, 6)
M2	(8, 12)		

Table 10
The processing information of the distributed manufacturing enterprise.

Job	Operation	F_1		F_2	
		Compatible machine	Processing time	Compatible machine	Processing time
J_1	O_{11}	1,2,3,4,5	13,10,8,10,11	1,2,3,4,5	6,10,9,7,15
	O_{12}	1,2,3	8,18,12	1,2,3	5,6,17
	O_{13}	1,2,3,5	12,13,16,19	1,2,3,5	15,10,14,14
	O_{14}	1,3,4,5	10,18,14,12	1,3,4,5	7,12,9,12
J_2	O_{21}	1,2,5	18,8,7	1,2,5	11,6,8
	O_{22}	2,3,4,5	16,17,14,19	2,3,4,5	8,18,12,14
	O_{23}	1,2,3,5	18,15,12,17	1,2,3,5	17,7,16,10
J_3	O_{31}	1,3,4	6,5,9	1,3,4	6,17,10
	O_{32}	1,4	6,19	1,4	15,9
	O_{33}	1,2,3,4,5	19,13,13,19,7	1,2,3,4,5	11,17,11,5,10
	O_{34}	1,2,3,4	12,11,15,17	1,2,3,4	10,7,14,10
J_4	O_{41}	1,3	7,8	1,3	6,15
	O_{42}	3,4,5	5,9,6	3,4,5	12,10,6
	O_{43}	1,3,4,5	17,6,12,8	1,3,4,5	13,17,19,13
	O_{44}	1,2,4,5	8,12,15,17	1,2,4,5	11,13,17,14
J_5	O_{51}	1,2,3,4,5	8,13,16,10,19	1,2,3,4,5	17,6,9,5,19
	O_{52}	1,4	19,11	1,4	12,9
	O_{53}	1,2,3,4,5	13,15,17,6,8	1,2,3,4,5	15,19,16,5,9
J_6	O_{61}	1,2,3,4,5	17,17,12,14,9	1,2,3,4,5	15,7,9,9,12
	O_{62}	1,2,4,5	14,13,15,18	1,2,4,5	14,18,14,13
	O_{63}	4,5	18,12	4,5	14,8
J_7	O_{71}	1,2,3,5	7,9,16,19	1,2,3,5	10,10,8,6
	O_{72}	1,2,4,5	11,16,15,14	1,2,4,5	9,12,6,19
	O_{73}	1,2,3,4	13,13,18,11	1,2,3,4	19,5,17,7
	O_{74}	1,2,3,4	7,11,14,18	1,2,3,4	10,12,5,14
J_8	O_{81}	1,2,4,5	11,19,16,5	1,2,4,5	9,7,8,18
	O_{82}	1,5	7,6	1,5	19,15
	O_{83}	3,5	16,9	3,5	19,10
J_9	O_{91}	1,3,4,5	10,13,5,14	1,3,4,5	16,8,19,19
	O_{92}	1,2,3,4,5	15,7,8,7,10	1,2,3,4,5	19,16,8,16,8
	O_{93}	1,4	13,9	1,4	13,9
J_{10}	O_{101}	4,5	19,16	4,5	6,18
	O_{102}	1,2,4,5	15,11,17,11	1,2,4,5	9,8,13,13
	O_{103}	1,2,3,4,5	12,9,15,15,14	1,2,3,4,5	19,10,18,18,19
J_{11}	O_{111}	1,2,3,4,5	12,12,6,15,19	1,2,3,4,5	12,15,10,15,13
	O_{112}	1,2,3,4,5	9,19,5,6,6	1,2,3,4,5	9,5,7,16,10
	O_{113}	1,2,3,4,5	7,18,8,7,19	1,2,3,4,5	9,7,10,10,18
	O_{114}	1,3,5	15,14,8	1,3,5	16,19,8
J_{12}	O_{121}	1,2,4,5	15,7,17,15	1,2,4,5	11,13,19,6
	O_{122}	1,2,3,4,5	17,10,6,17,13	1,2,3,4,5	10,15,16,14,11
	O_{123}	2,3,5	17,18,16	2,3,5	19,11,17
	O_{124}	3,4	12,16	3,4	10,6
J_{13}	O_{131}	1,2,3,4,5	12,16,15,12,9	1,2,3,4,5	18,19,8,18,16
	O_{132}	1,2,3,4,5	5,17,16,18,13	1,2,3,4,5	5,10,11,13,9
	O_{133}	1,2,3,4,5	18,10,6,18,16	1,2,3,4,5	5,16,16,7,10
	O_{134}	1,2,3,5	13,8,14,7	1,2,3,5	17,8,8,6
J_{14}	O_{141}	2,4	8,7	2,4	17,13
	O_{142}	1,2,3,4,5	19,15,5,13,11	1,2,3,4,5	14,10,8,6,17
	O_{143}	2,3,4,5	13,6,11,16	2,3,4,5	15,15,15,18
J_{15}	O_{151}	1,2,3,4,5	16,13,14,12,11	1,2,3,4,5	19,15,5,9,10
	O_{152}	2,4,5	19,13,18	2,4,5	5,13,5
	O_{153}	2,3	6,13	2,3	6,11
	O_{154}	1,2,3	18,9,9	1,2,3	11,12,11
J_{16}	O_{161}	1,3,4,5	10,8,10,18	1,3,4,5	12,19,13,17
	O_{162}	4,5	13,6	4,5	16,17
	O_{163}	2,4	14,7	2,4	15,10
	O_{164}	3,5	14,7	3,5	8,7
J_{17}	O_{171}	1,2,4,5	16,11,11,18	1,2,4,5	14,8,17,17
	O_{172}	1,2,3,4	9,8,12,13	1,2,3,4	5,15,12,15
	O_{173}	1,2,4,5	11,11,12,14	1,2,4,5	9,18,11,15
	O_{174}	2,3,4	5,12,15	2,3,4	10,12,9
J_{18}	O_{181}	1,2,3,4,5	16,14,16,17,18	1,2,3,4,5	10,18,19,16,7
	O_{182}	1,3,5	9,13,14	1,3,5	10,7,13
	O_{183}	1,2,3,4,5	19,6,16,7,12	1,2,3,4,5	13,8,8,13,13
J_{19}	O_{191}	1,2,3,4,5	11,5,14,11,18	1,2,3,4,5	17,15,5,8,9
	O_{192}	1,2	14,6	1,2	15,14
	O_{193}	4,5	14,5	4,5	17,17
	O_{194}	2,3,5	11,17,9	2,3,5	15,12,6
J_{20}	O_{201}	1,2,3,4	16,5,9,17	1,2,3,4	16,10,15,13
	O_{202}	1,2,3,4,5	6,9,14,5,8	1,2,3,4,5	10,14,11,14,16
	O_{203}	2,3,4,5	11,18,17,9	2,3,4,5	19,7,17,16
	O_{204}	2,5	12,8	2,5	18,11

Experiment Results--Example verification

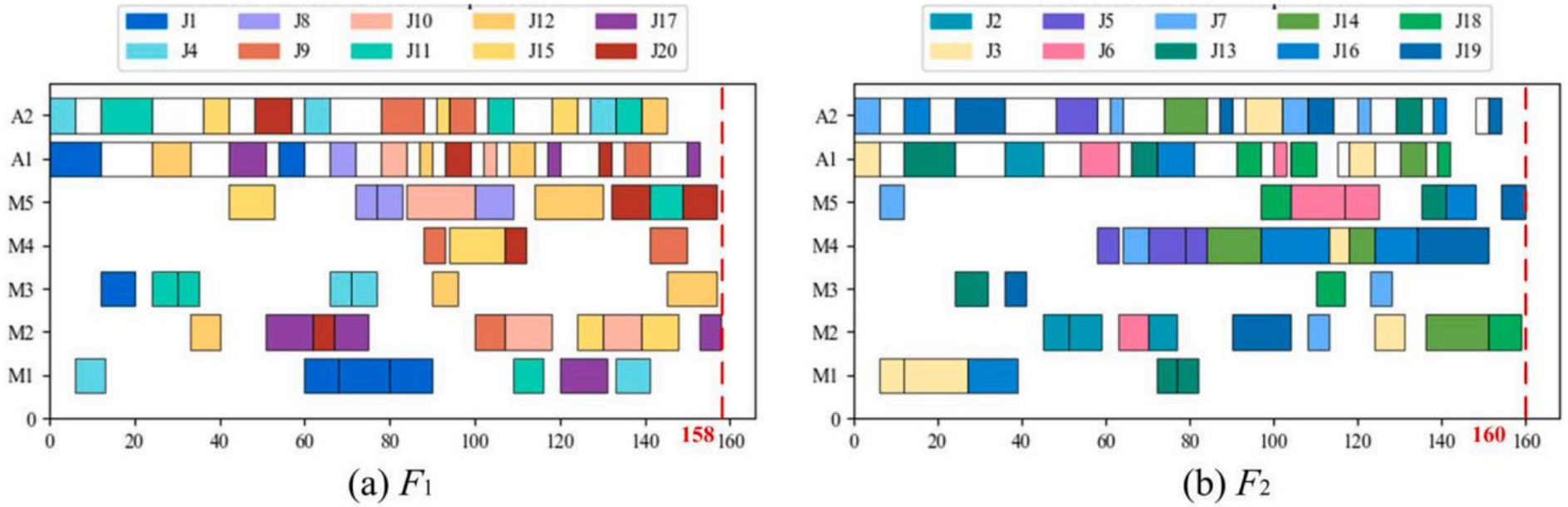


Fig. 11. Scheduling Gantt chart of the distributed manufacturing enterprise.

Contents

Future Work

Future work could explore more advanced state representation techniques to enhance decision-making accuracy. As the problem scale increases, the expanded state-action space may demand higher computational resources. Approaches such as distributed training, hierarchical modeling, and attention mechanisms could be employed to address large-scale environments more efficiently. Incorporating multi-objective optimization in future research could further enhance the practical applicability of the proposed method.

Paper 4

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

Energy-efficient Distributed Heterogeneous Hybrid Flow-shop Scheduling Using Graph Neural Network and Deep Reinforcement Learning

Haizhu Bao, Quanke Pan, Chee-Meng Chew, Ling Wang, and Liang Gao

Publish Journal: IEEE Transactions on Automation Science and Engineering

Published Time: 23 July 2025

IF: 6.4

Contents

Contents

Problem Description

This paper tackles the energy-efficient distributed heterogeneous hybrid flow-shop scheduling problem (EDHHFSP), aiming to minimize both makespan and total energy consumption.

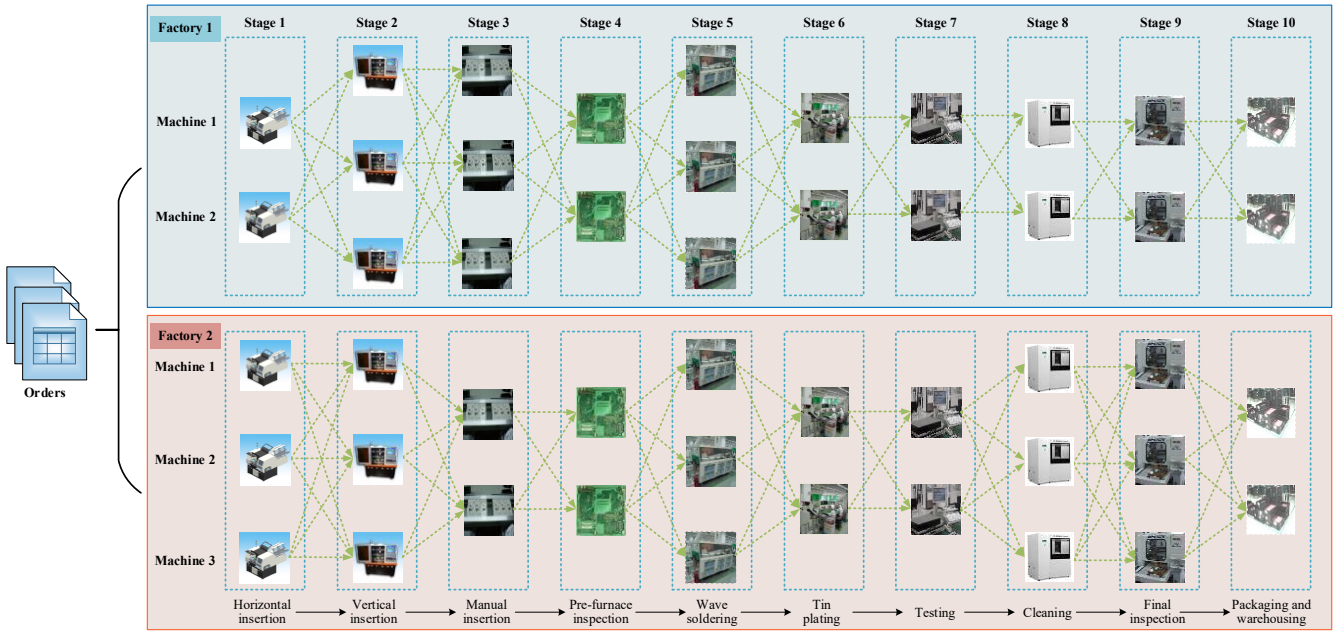
We first formulate a mixed-integer linear programming (MILP) model to provide a benchmark for small instances. More importantly, we propose a novel end-to-end deep reinforcement learning framework based on a heterogeneous graph neural network, which models the scheduling problem as a distributed decision-making process. A key innovation lies in the design of an action space composed of "job–factory" and "operation–machine" pairs, enabling fine-grained, decentralized scheduling decisions.

Our approach starts with a novel heterogeneous graph representation of scheduling states, capturing complex interactions among jobs, factories, and machines. A three-stage embedding mechanism is developed to encode real-time scheduling environments. The agent then learns a parameterized policy using the proximal policy optimization (PPO) algorithm, guided by a reward function that balances makespan and energy efficiency.

Contents

Problem Description

The EDHHFSP comprises δ heterogeneous factories, each characterized by distinct processing capacities. Each factory operates as a flexible flow-shop with m sequential stages. Given the diversity in flexibility across different factories, each factory F_f is equipped with $l_{f,k}$ unrelated parallel machines possessing different technology levels at stage K_k. Denoted as $M_i^{f,k}$, the ith machine at stage K_k in F_f has a processing speed $v_{f,k,i}$. A set of n jobs must be allocated to one of the δ factories and subsequently executed on any of the UPMs across m stages sequentially within a specific factory. Each job J_j consists of m operations $(O_{j,1}, O_{j,2}, \ldots, O_{j,m})$, where the standard time for $O_{j,k}$ is $p_{j,k}$. The actual processing time of $O_{j,k}$ on $M_i^{f,k}$ is $p_{j,k}/v_{f,k,i}$, with an EC per unit time of $\beta_{f,i,k}$. Due to the lack of intermediate buffers between parallel machines at different stages, the current machine will retain the job until a machine in the next stage is idle, resulting in a blocking effect on the current machine. Additionally, in alignment with practical production scenarios, setup times are distinct from the processing times and depend on the job sequence. Thus, each machine $M_i^{f,k}$ can operate under four distinct EC modes, namely processing mode EC ($PEC_{j,i,f,k}$), setup mode EC ($SEC_{j,i,f,k}$), blocking mode EC ($BEC_{j,i,f,k}$), and idle mode EC ($IEC_{j,i,f,k}$). The EDHHFSP seeks to minimize C_{max} and TEC, while also incorporating other typical flow-shop constraints. (1) All jobs become available at time zero; (2) each job is assigned to a single machine at any given time; (3) each machine handles only one job at a time; (4) preemption is strictly prohibited.



Method Design

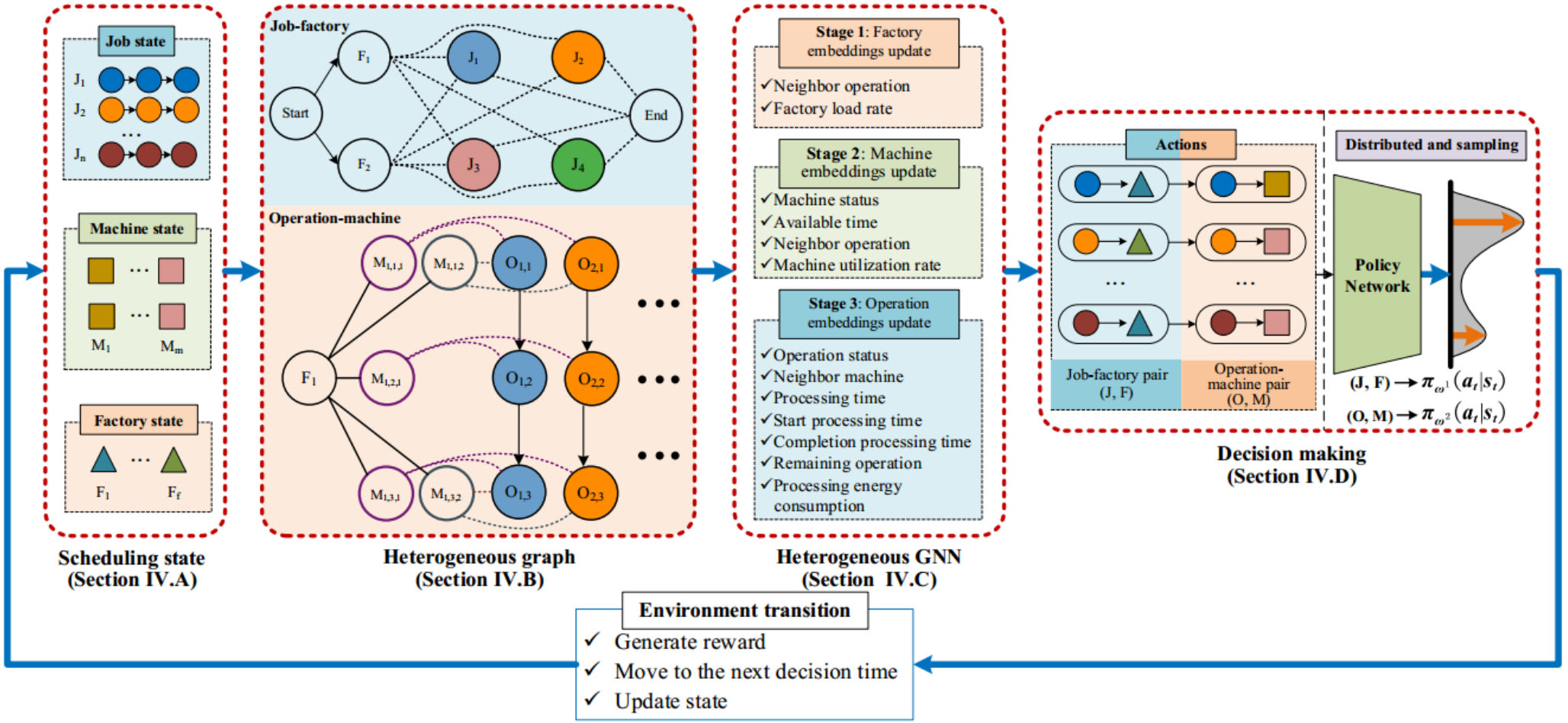


Fig. 1. Workflow of the proposed methodology for EDHHFSP.

MDP--State representation (Heterogeneous Graph)

$$\mathcal{H}^D = (\mathcal{F}, \mathcal{O}, \mathcal{M}, \mathcal{C}, \mathcal{E}, \mathcal{K})$$

factory nodes $(\mathcal{F})$

operation nodes $(\mathcal{O})$

machine nodes $(\mathcal{M})$

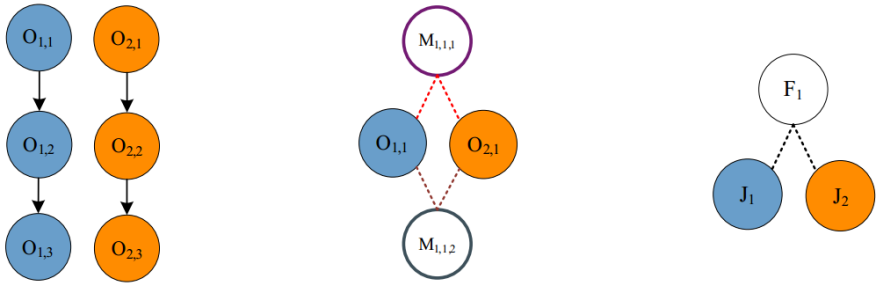


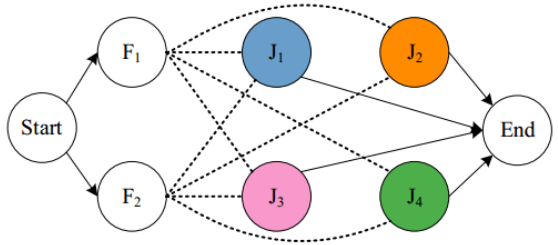
Fig. 2. Illustration of arc types in the scheduling model: (a) Conjunctive arcs: representing fixed sequential dependencies between operations, (b) disjunctive arcs between operations and machines: highlighting alternative paths for operation-to-machine assignments, (c) disjunctive arcs between jobs and factories: showing flexible allocation options for assigning jobs to factories.

conjunctive arcs to enforce processing sequence constraints $(\mathcal{C})$

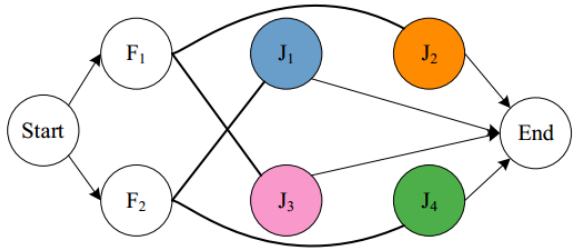
disjunctive arcs to allocate operations to machines $(\mathcal{E})$

additional disjunctive arcs to define allocation relationships between jobs and factories $(\mathcal{K})$

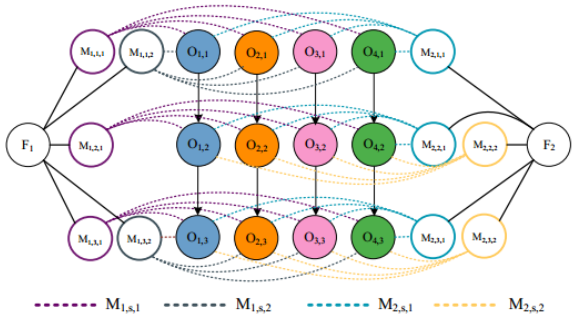
MDP--State representation (Heterogeneous Graph)



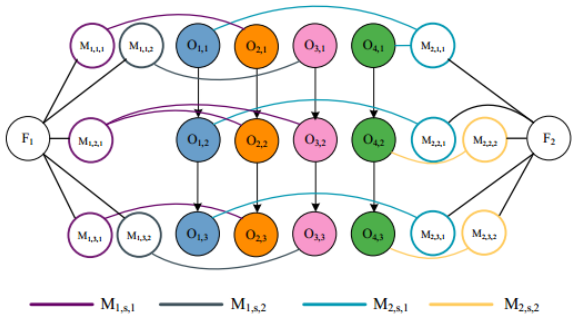
(a) An instance of factory assigning.



(c) A solution of factory assigning.



(b) An instance of machine assigning.



(d) A solution of machine assigning.

Fig. 3. Heterogeneous graph representation of EDHHFSP. Dashed lines represent processable relationships, indicating potential operations or assignments, while solid lines denote scheduled connections, reflecting confirmed operations or allocations within the scheduling framework.

MDP--Lower-level Agent

Action

job-factory pairs (J, F)

operation-machine pairs (O, M)

Reward

The reward function integrates both the action space and the state representation, guiding the optimization of the strategy [13], [69]. The reward function is defined as Eq. (24), where w_1 and w_2 are weights, ensuring that the magnitudes of the first and second terms are comparable. When the discount factor $\gamma = 1$, the cumulative reward is calculated as

$$r(s_t, a_t, s_{t+1}) = w_1(C_{max}(s_t) - C_{max}(s_{t+1})) + w_2(TEC(s_t) - TEC(s_{t+1})) \quad (24)$$

$$G = \sum_{t=0}^{|O|} r(s_t, a_t, s_{t+1}) = w_1(C_{max}(s_0) - C_{max}) + w_2(TEC(s_0) - TEC). \quad (25)$$

For a given problem instance, $C_{max}(s_0)$ and $TEC(s_0)$ represent the C_{max} and TEC values of the environment in the initial state s_0, respectively, and they are both estimated constants. The method details are as follows: $C_{max}(s_0) = \max_{j \in J} CT_j$, $TEC(s_0) = \sum_{j \in J, k \in K} ec_{j,k}$. Therefore, maximizing the cumulative reward G aligns with minimizing C_{max} and TEC, which constitutes the primary objective of this paper.

Heterogeneous Graph Neural Network

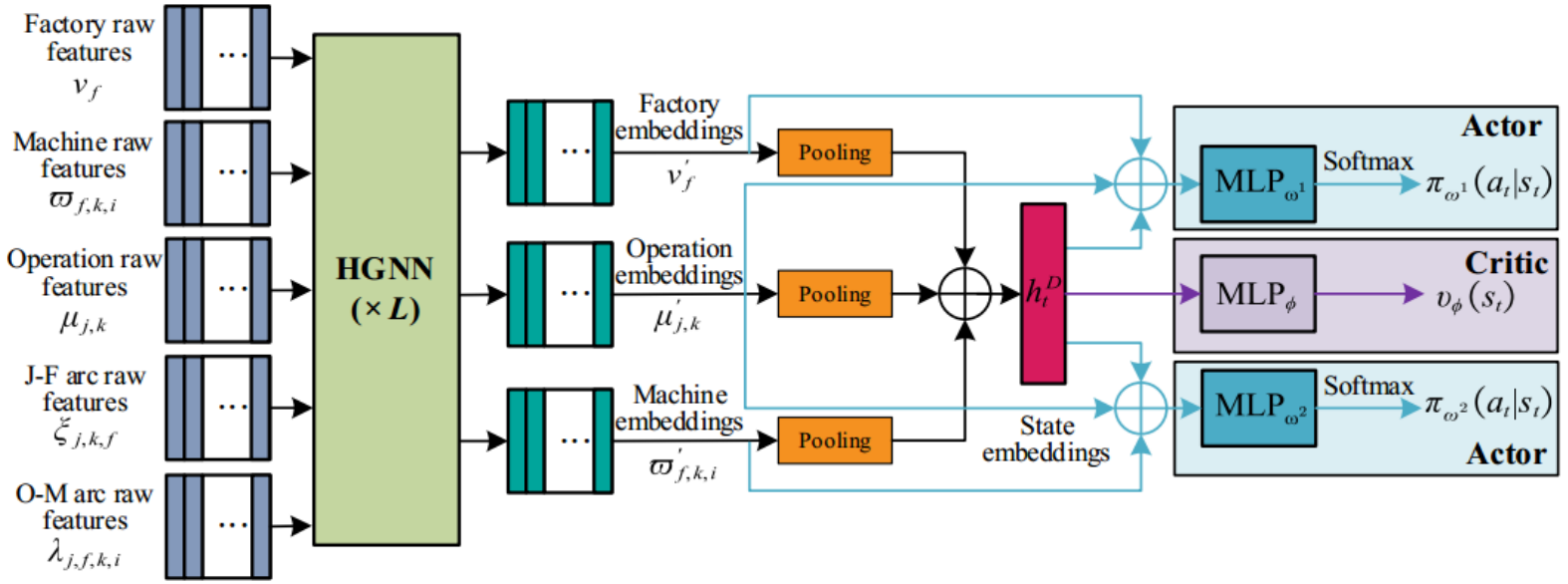


Fig. 4. The network architecture.

Factory node embedding	Machine node embedding	Operation node embedding
GAT	GAT	MLP

Contents

Experiment Results——Evaluation instances and Comparison methods

TABLE S-VI
FACTORS AND LEVELS

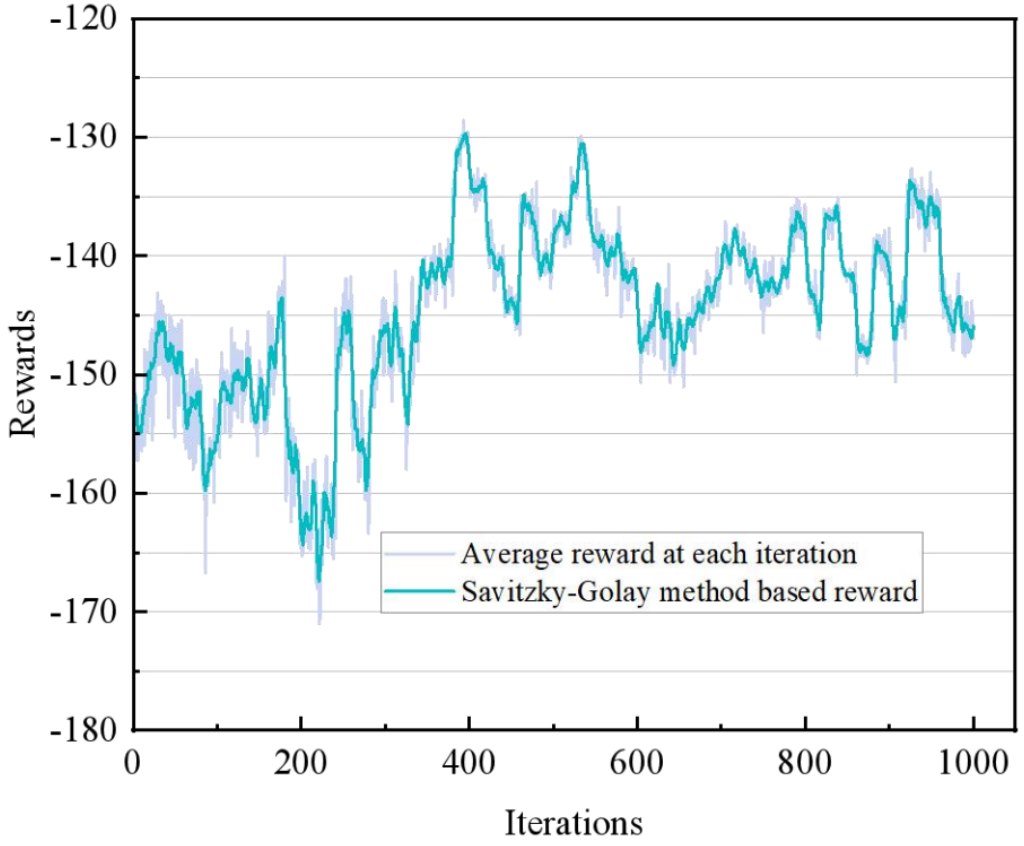
Factors	Notations	Levels
Number of factories	δ	$\delta \in \{2,3,4,5,6\}$
Number of jobs	n	$n \in \{15,20,40,60\}$
Number of stages	m	$m \in \{5,10\}$
Number of machines at K_k in F_f	$l_{f,k}$	$l_{f,k} \sim U[2,5]$
Standard processing time of J_j at K_k	$p_{j,k}$	$p_{j,k} \sim U[10,50]$
Setup time for changeover from J_j to $J_{j'}$ on M_i at K_k	$st_{j,j',i,k}$	$st_{j,j',i,k} \sim U[10,20]$
Speed of $M_{f,k,i}$	$v_{f,k,i}$	$v_{f,k,i} \in \{1,1.1,1.2,1.3,1.4\}$
EC per unit time of $M_{f,k,i}$ at processing mode	$\beta_{f,i,k}$	$\beta_{f,i,k} \sim U[5,10]$
EC per unit time at setup mode	γ	$\gamma = 1$
EC per unit time at blocked mode	ω	$\omega = 0.7$
EC per unit time at idle mode	θ	$\theta = 0.5$

We compare the proposed method with several existing approaches:

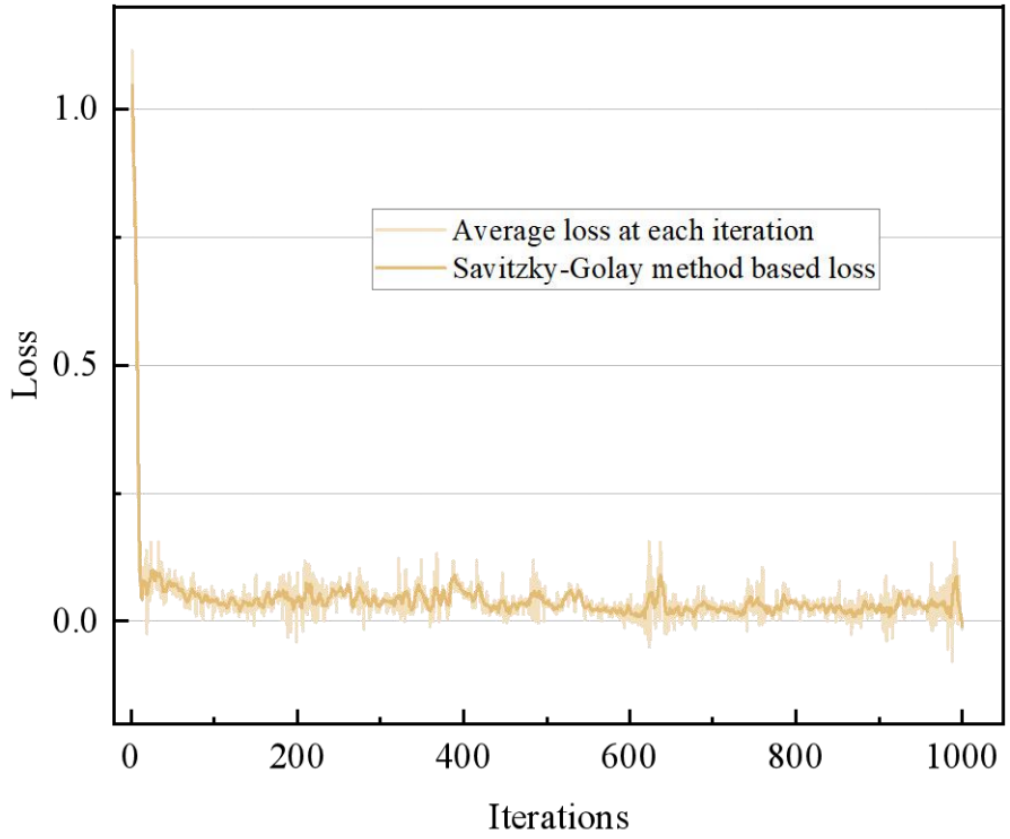
- PDRs: longest processing time (LPT), shortest processing time (SPT);
- Metaheuristic algorithms: HAS [30], MMMA [8], QABC [2];
- Multi-objective optimization method: NSGA-II [6];
- DRL-based method: DRL-FJSP [13], DRLCEA [53].

We have redesigned these comparative algorithms to adapt to EDHHFSP. Please refer to the supplementary materials for more information. For fair comparisons, the termination criteria of HAS, MMMA, NSGA-II, QABC, and DRLCEA are based on the maximum elapsed CPU time ($c \times n \times m \times \delta$) milliseconds, with the c set to 20 [7]. Hypervolume (HV) and inverse generational distance (IGD) metrics are considered in this paper.

Experiment Results——The convergence performance analysis of DRL



(a) Reward



(b) Loss

Fig. S-7. Training process visualization of $n = 20, m = 5, \delta = 4$.

Experiment Results--Evaluation of instances of training size

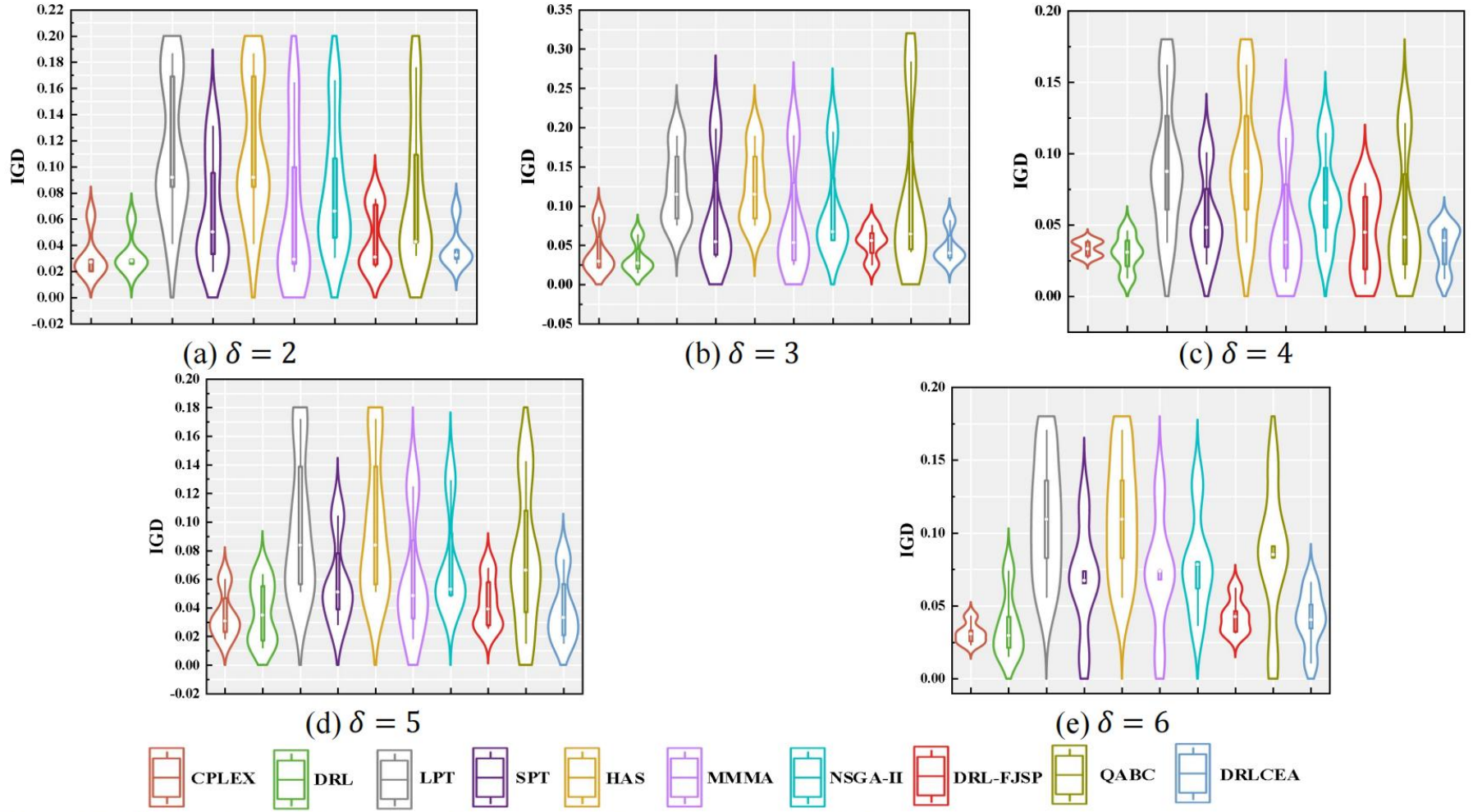


Fig. S-8. Violin plots of the average IGD on training instances of all algorithms.

Experiment Results——Ablation studies

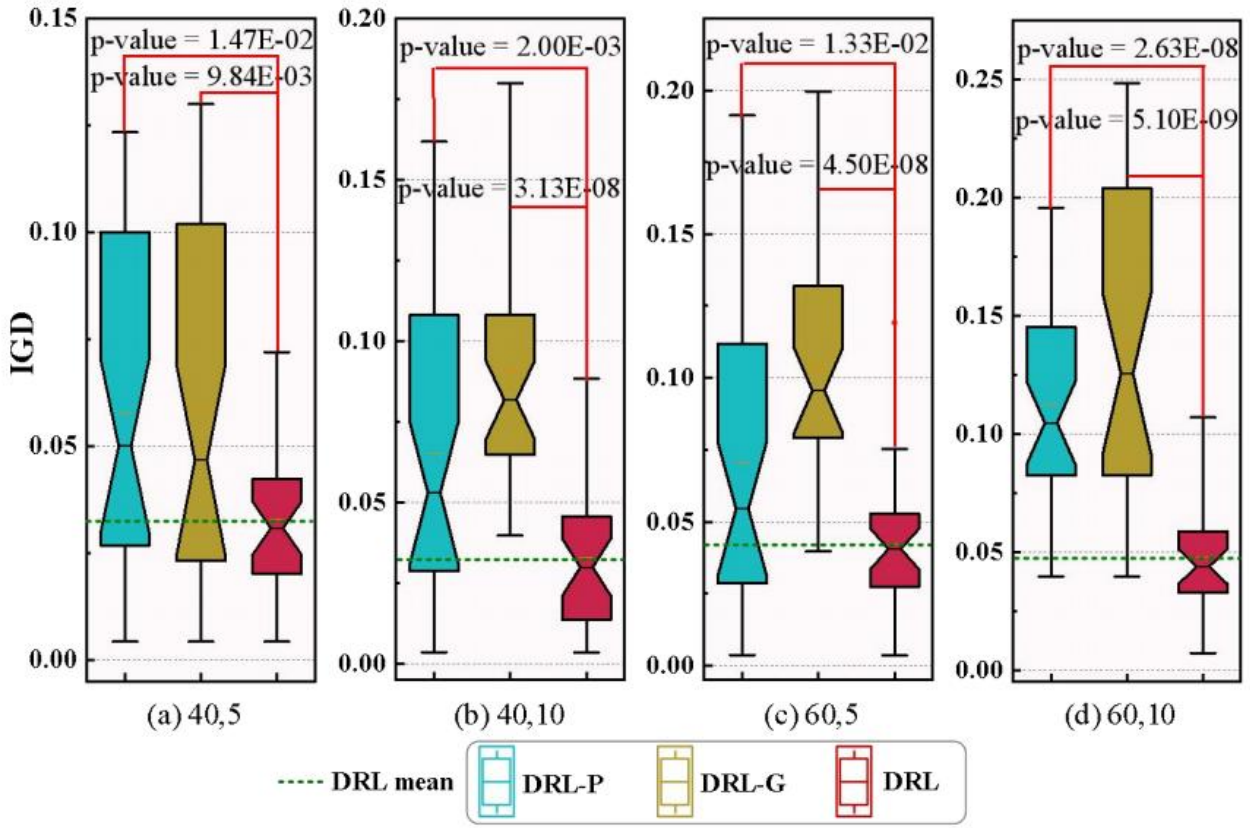


Fig. S-9. Boxplots of IGD values of ablation study and one-way ANOVA.

Experiment Results——Generalization performance on large-sized instances

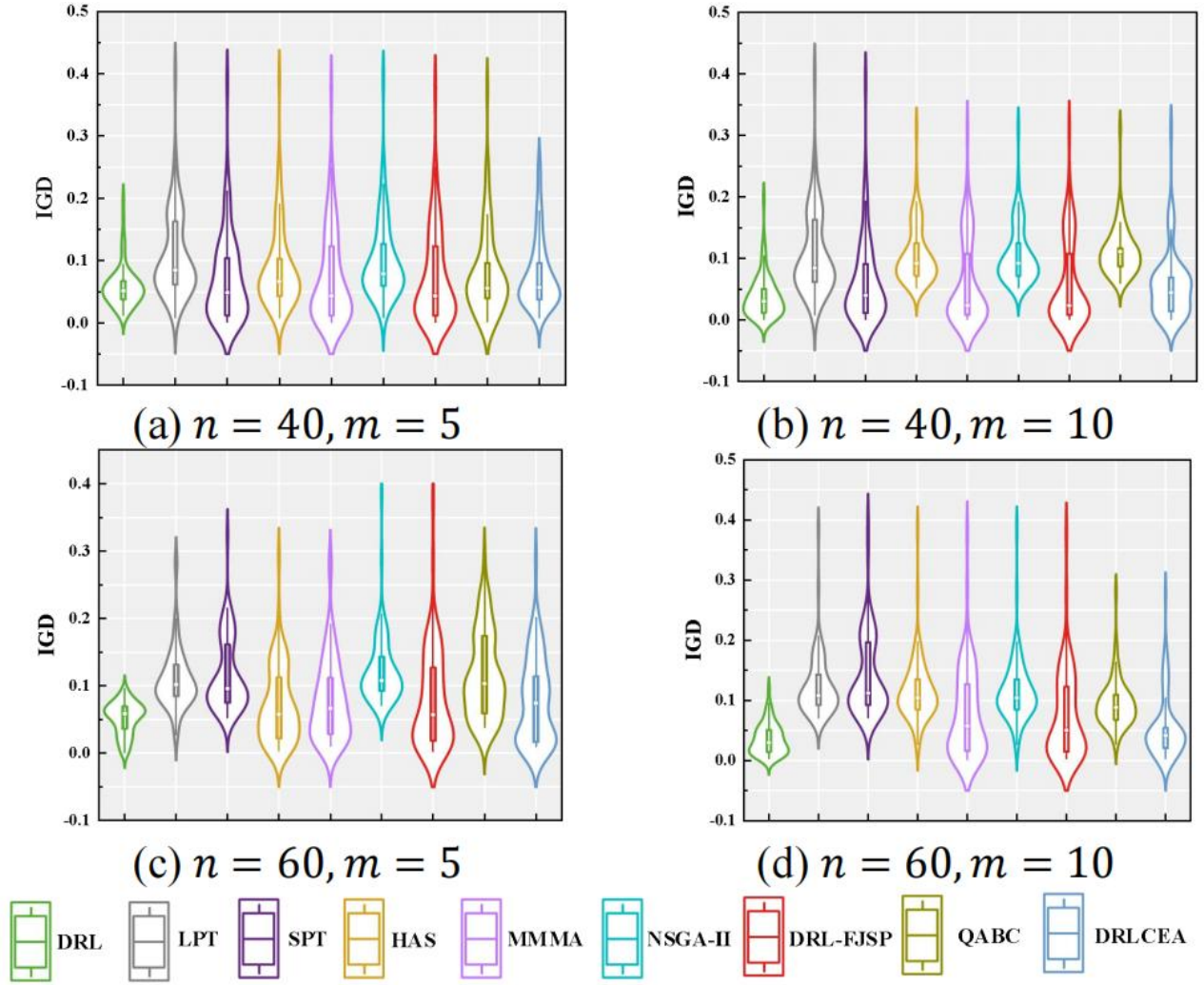


Fig. S-10. Violin plots of the average IGD on large-sized instances of all algorithms.

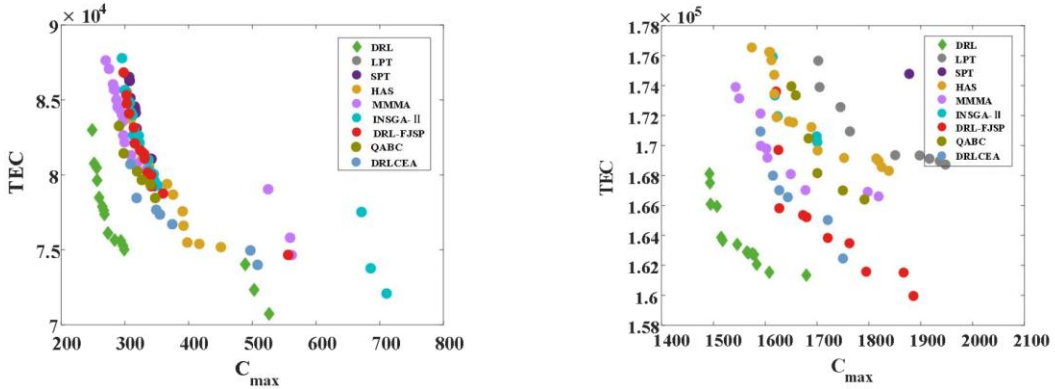
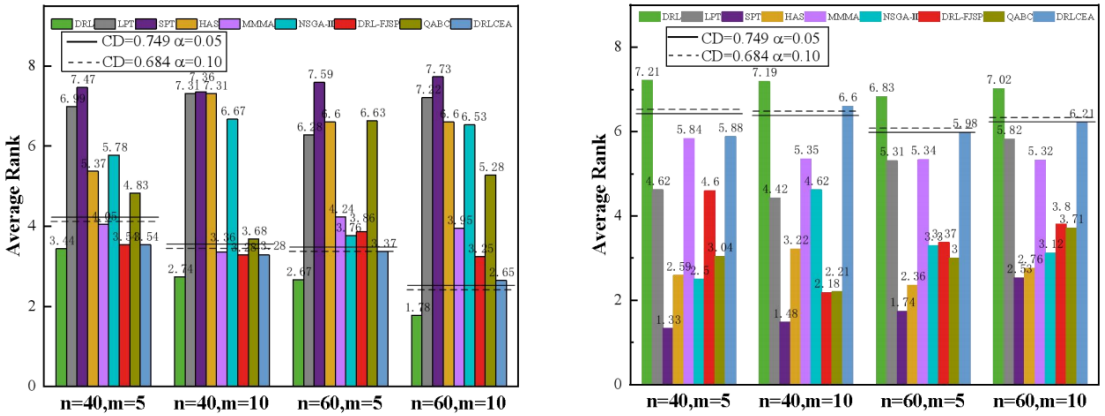


Fig. S-11. Pareto frontiers on some large-sized instances.

Experiment Results——Generalization performance on large-sized instances

TABLE I
RESULTS ACHIEVED BY FRIEDMAN TEST

Algorithms	$n = 40, m = 5$		$n = 40, m = 10$		$n = 60, m = 5$		$n = 60, m = 10$	
	IGD	HV	IGD	HV	IGD	HV	IGD	HV
DRL	3.44	7.21	2.74	7.19	2.67	6.83	1.78	7.02
LPT	6.99	4.62	7.31	4.42	6.28	5.31	7.22	5.82
SPT	7.47	1.33	7.36	1.48	7.59	1.74	7.73	2.53
HAS	5.37	2.59	7.31	3.22	6.60	2.36	6.60	2.76
MMMA	4.05	5.84	3.36	5.35	4.24	5.34	3.95	5.32
NSGA-II	5.78	2.50	6.67	4.62	3.76	3.30	6.53	3.12
DRL-FJSP	3.54	4.60	3.28	2.18	3.86	3.37	3.25	3.80
QABC	4.83	3.04	3.68	2.21	6.63	3.00	5.28	3.71
DRLCEA	3.54	5.88	3.28	6.60	3.37	5.98	2.65	6.21
CN	200	200	200	200	200	200	200	200
p-value	5.05E-05	4.22E-05	5.08E-05	2.06E-04	4.00E-04	3.21E-04	2.07E-04	4.25E-04
Crit. Diff $\alpha = 0.05$					0.749			
Crit. Diff $\alpha = 0.1$					0.684			



(a) IGD (b) HV

Fig. S-12. Friedman-test at different metrics of all algorithms.

Experiment Results——Run time analysis

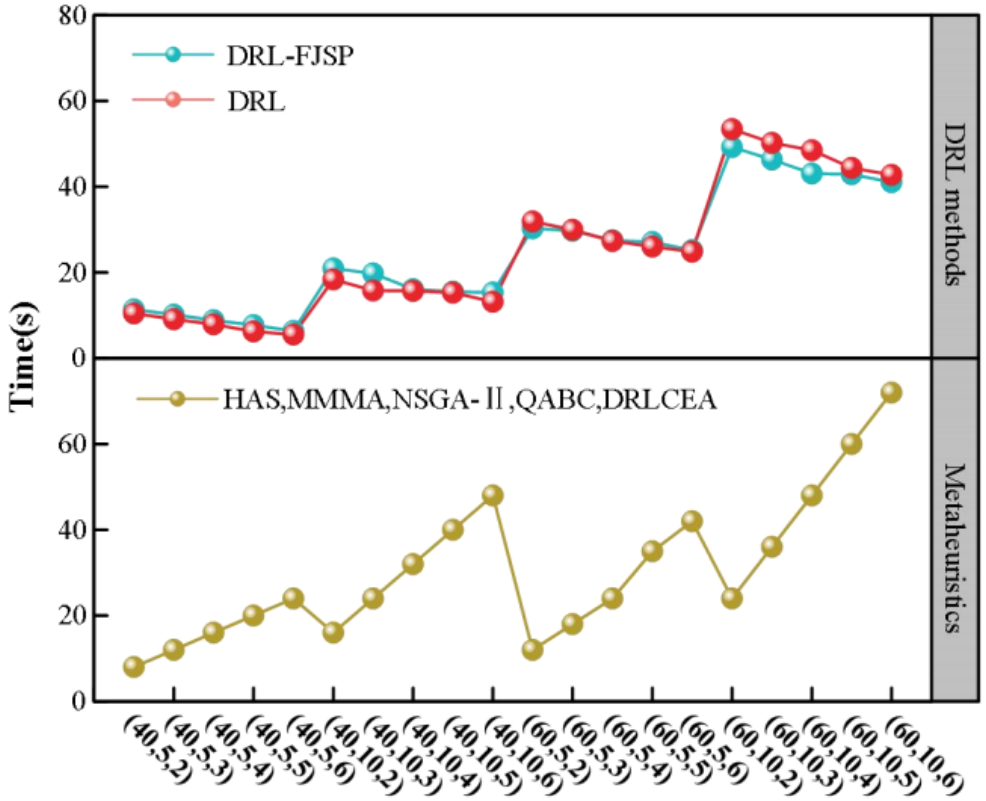


Fig. S-13. Average running time of algorithms.

Experiment Results--Industrial Case Study

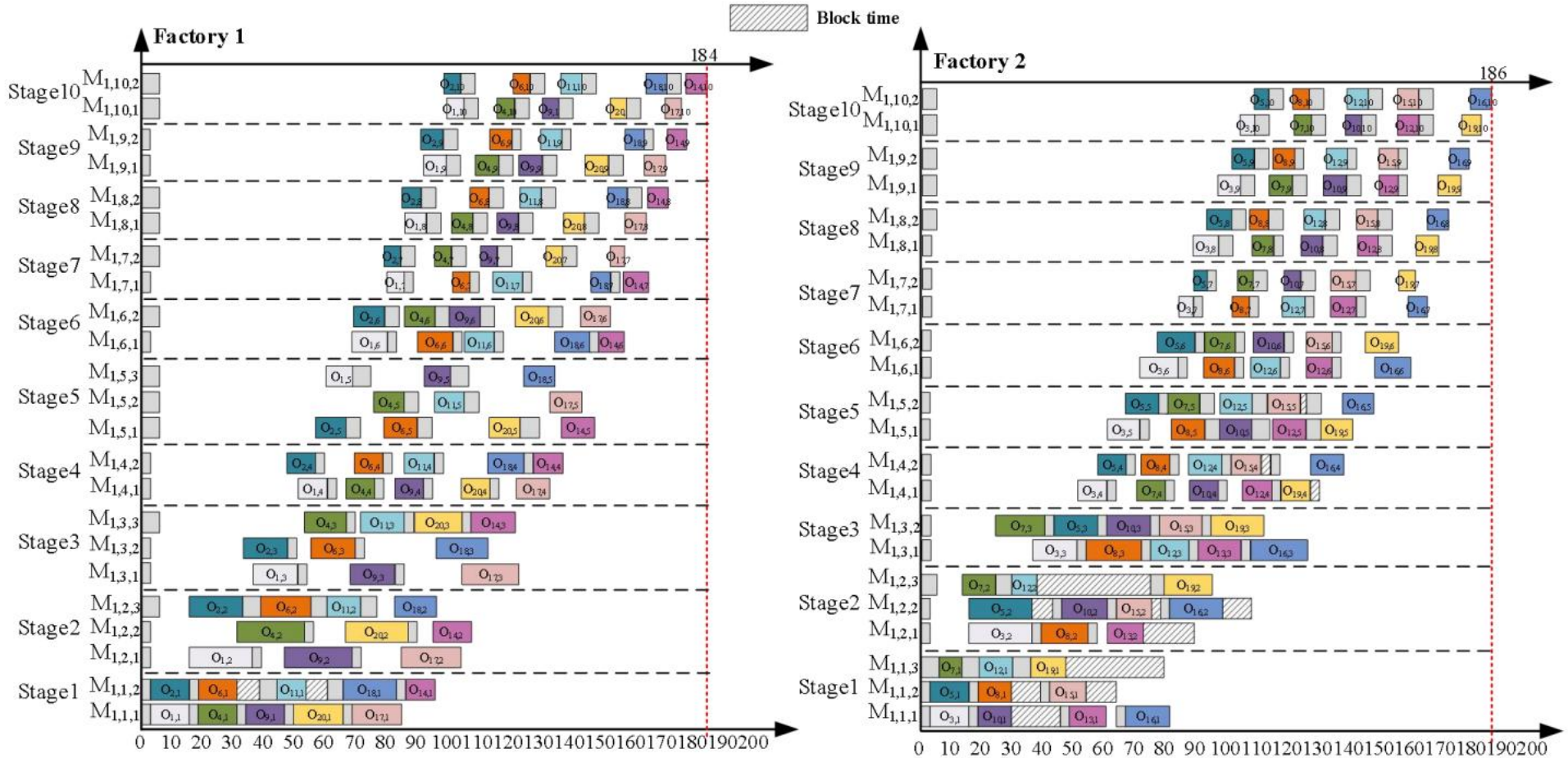


Fig. S-14. Gantt chart representing the scheduling scheme for the test case.

Contents

Future Work

Future work aims to incorporate additional real-world constraints into the environment, such as transportation times of workpieces between machines, batching processes, and system uncertainties including order cancellations and emergency orders.

Furthermore, the end-to-end DRL approach should be refined to better extract scheduling state features and integrate advanced search mechanisms, ultimately improving training performance and robustness.

Paper 5

北京工業大學
BEIJING UNIVERSITY OF TECHNOLOGY

An End-to-End Framework for Energy-Efficient Cascaded Dual-Shop Collaborative Scheduling With Mating Operations

Lixiang Zhang , Chen Yang , Member, IEEE, Yan Yan , and Yaoguang Hu

Publish Journal: IEEE TRANSACTIONS ON CYBERNETICS

Published Time: 25 July 2025

IF: 10.5

Contents

Contents

Problem Description

This article investigates a collaborative scheduling problem in a cascaded dual-shop production setting. Unlike singleshop scheduling or distributed multiworkshop scheduling, this problem emphasizes collaborative optimization between two interdependent workshops. In addition, real-world production often involves a mode where main and suborders must be integrated through mating operations.

This study formulates an energy-efficient cascaded dual-shop collaborative scheduling problem with the mating operation (ECDCSP-M). The focus is on developing a mixed-integer linear programming (MILP) model for the ECDCSP-M and designing an end-to-end graph-based deep reinforcement learning (GDRL) approach. A dual-shop heterogeneous graph is constructed to capture the real-time state of the entire system, in which "job-factory" and "operation-machine" pairs are defined as agent actions. A heterogeneous graph neural network (HGNN) is then proposed, employing a three-stage embedding mechanism to model complex relationships, including mating operations.

Contents

Method Design

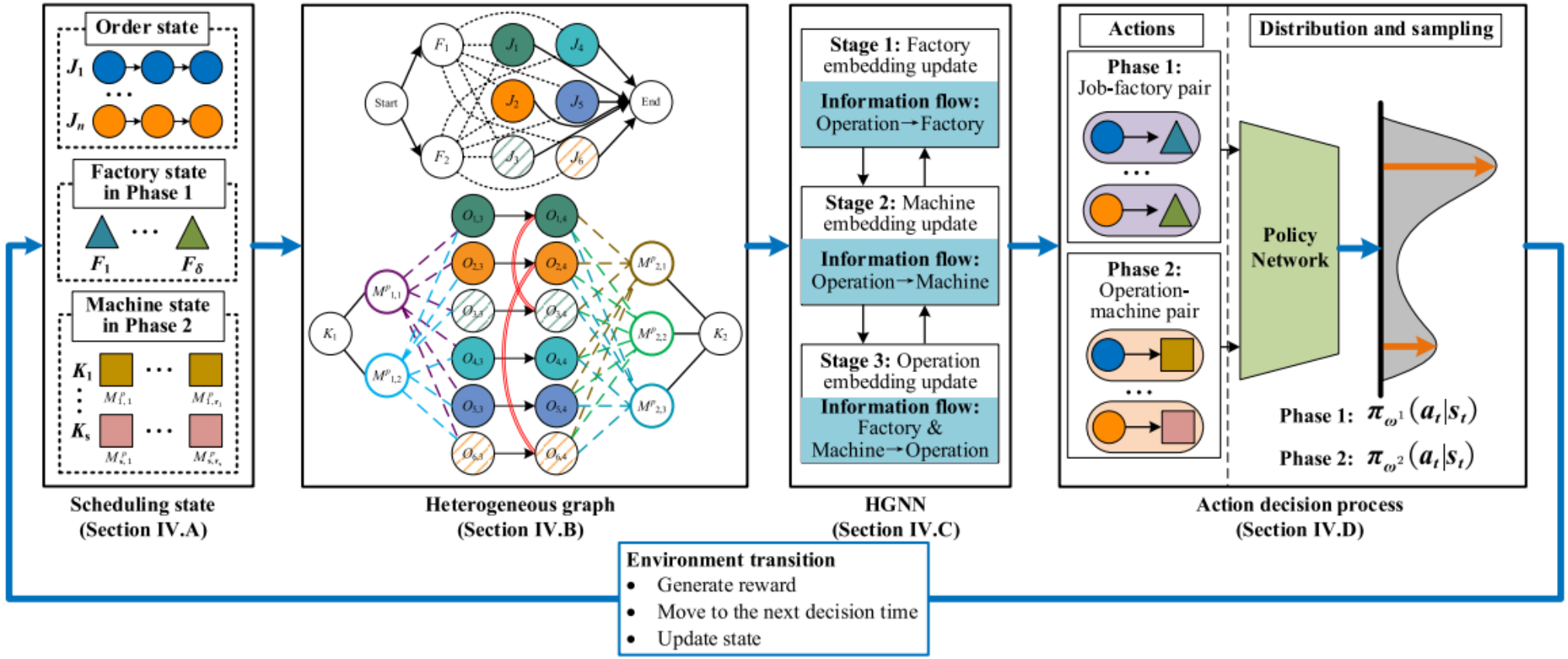


Fig. 4. Procedure of the proposed methodology.

Problem Description

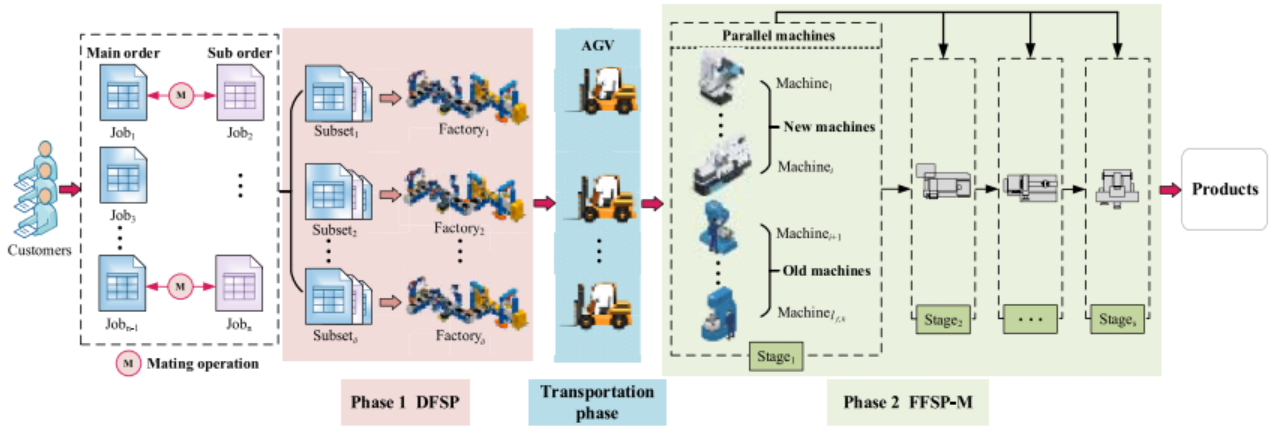


Fig. 1. Schematic view of the considered ECDCSP-M.

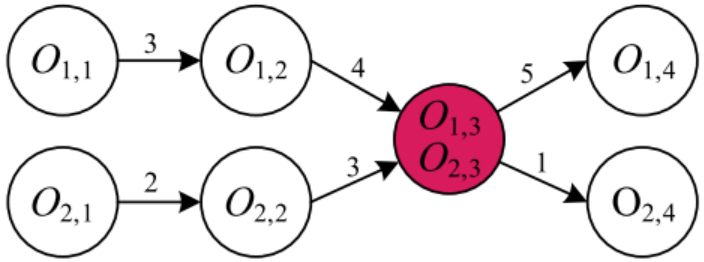


Fig. 2. Example of the precedence constraints in jobs modeled as DG.

The ECDCSP-M is formulated as a collaborative production process comprising two cascading flow shops. As illustrated in Fig. 1, the first phase (Phase 1) constitutes a DFSP, while the second phase (Phase 2) corresponds to a heterogeneous hybrid flow-shop scheduling problem with mating operations (HHSP-M). In Phase 1, δ identical factories are considered, each equipped with m machines. Each job can be assigned to any factory for processing. Within each factory, all jobs follow flow-shop permutation constraints, i.e., they must be processed in the same sequence across machines (from $M_{f,1}$ to $M_{f,m}$). Machine setup times are sequence-dependent but independent of processing durations. Upon completion in Phase 1, all n jobs are transported to Phase 2 for subsequent processing.

Problem Description

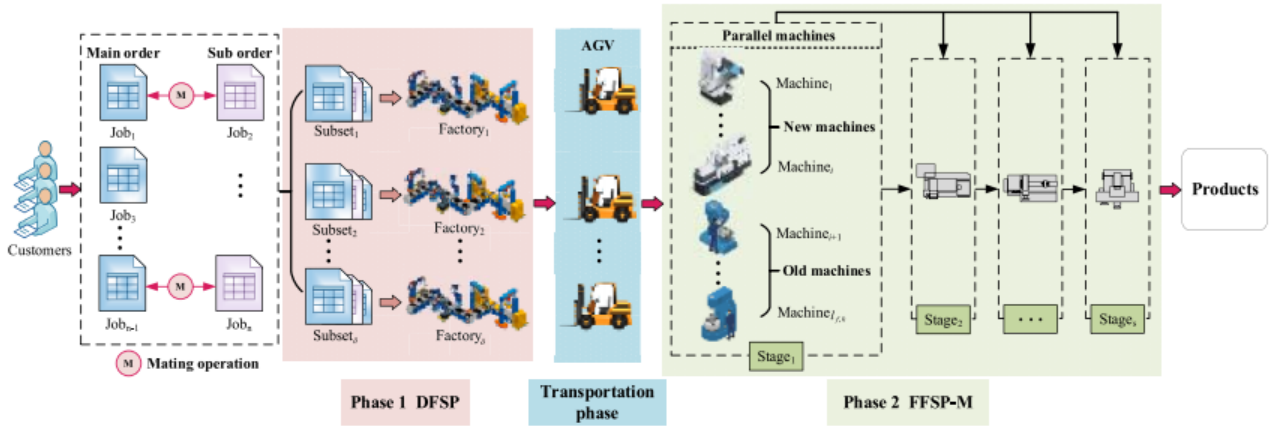


Fig. 1. Schematic view of the considered ECDCSP-M.

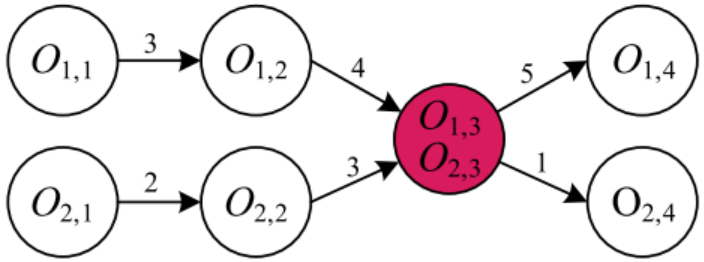


Fig. 2. Example of the precedence constraints in jobs modeled as DG.

In Phase 2, there are s production stages, along with all jobs follow a common production route, starting from stage K_1, and proceeding sequentially to K_s. Each stage comprises r_k unrelated parallel machines, including both new and legacy equipment, which differ in processing speeds $v_{k,l}$ and unit-time energy consumption (EC) $\beta^2_{k,l}$. Each job can be scheduled on any of the parallel machines at a given stage. To illustrate the mating operation more effectively, we introduce a directed graph (DG) model. As shown in Fig. 2, the DAG comprises Jobs 1 and 2, in which operations $O_{1,3}$ and $O_{2,3}$ must be executed simultaneously. In addition, the numbers on the edge represent the sequence-dependent setup times.

MDP--State representation (Heterogeneous Graph)

$$\mathcal{H}^D = (\mathcal{F}, \mathcal{O}, \mathcal{M}, \mathcal{C}, \mathcal{E}, \mathcal{K})$$

factory nodes $(\mathcal{F})$

operation nodes $(\mathcal{O})$

machine nodes $(\mathcal{M})$

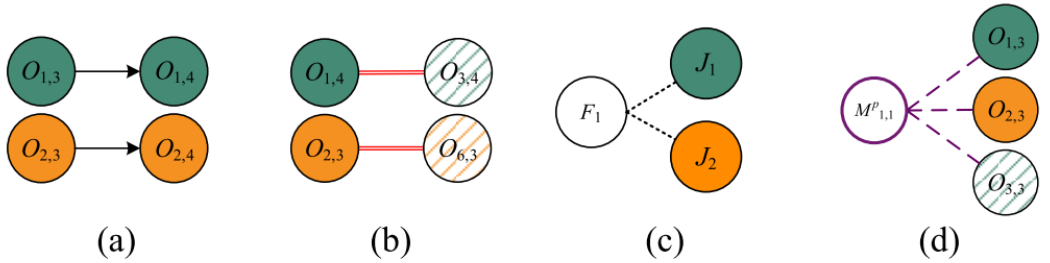


Fig. 5. Four types of arcs. (a) Conjunctive arcs represent operation dependencies. (b) Conjunctive arcs represent mating operations. (c) Disjunctive arcs represent the allocation of factories to jobs. (d) Disjunctive arcs represent the allocation of machines to operations.

conjunctive arcs to enforce processing sequence constraints $(\mathcal{C})$

disjunctive arcs to allocate operations to machines $(\mathcal{E})$

additional disjunctive arcs to define allocation relationships between jobs and factories $(\mathcal{K})$

MDP--State representation (Heterogeneous Graph)

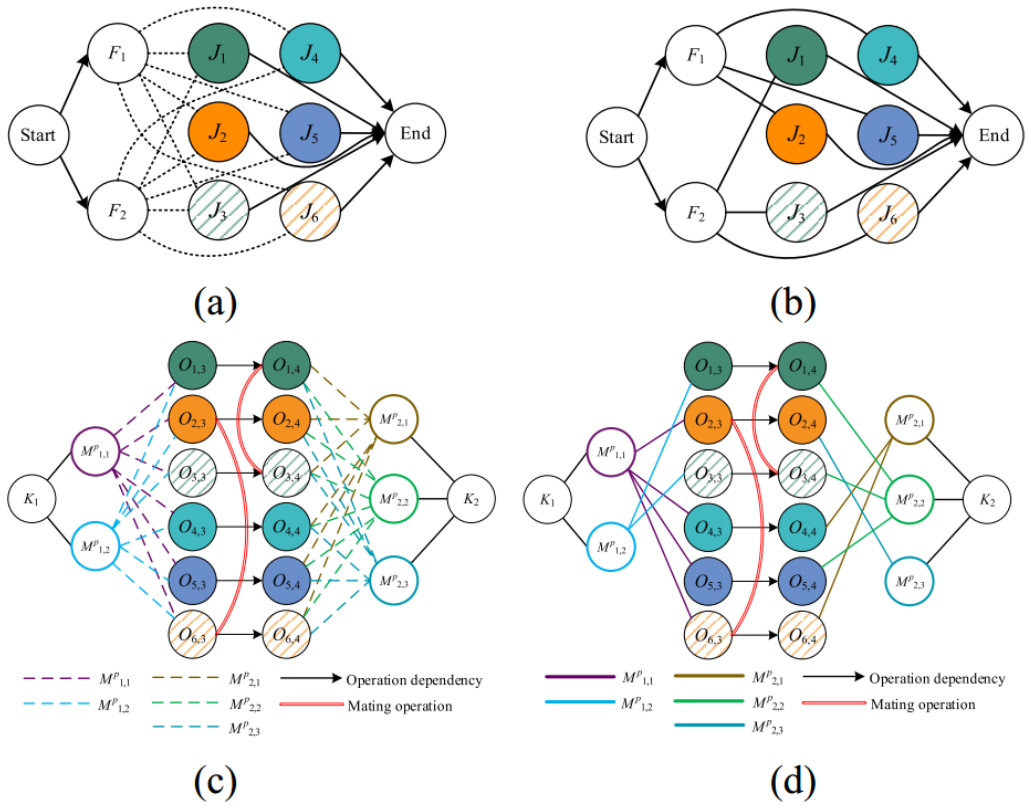


Fig. 6. Heterogeneous graph of ECDCSP-M. The dashed line means processable, while the solid line means scheduled (a) Instance of job assigning in Phase 1 (b) Instance of machine assigning in Phase 2. (c) Solution of job assigning in Phase 1. (d) Solution of machine assigning in Phase 2.

To address these issues, this article proposes a novel disjunctive graph $\mathcal{G}^H = (\mathcal{F}, \mathcal{M}, \mathcal{O}, \mathcal{C}, \mathcal{H}, \mathcal{K}, \mathcal{E})$, where $\mathcal{F}$ denotes additional factory nodes, $\mathcal{K}$ is the set of arcs connecting jobs and factories in Phase 1, and $\mathcal{H}$ represents arcs for mating operations in Phase 2. The remaining symbols are consistent with those used in previously discussed graphs [21]. As illustrated in Fig. 6, the traditional arcs $\mathcal{D}$ are replaced by the operation-machine pairs (O, M) arcs set $\mathcal{E}$, where each element $E_{j,k,l} \in \mathcal{E}$ is an undirected arc linking an operation node $O_{j,k}$ to a compatible machine node $M^p_{k,l}$ in Phase 2. Furthermore, factory nodes $\mathcal{F}$ and an undirected arc set $\mathcal{K}$, representing job-factory pairs (J, F), are incorporated. Each element $K_{j,f} \in \mathcal{K}$ is an undirected arc linking a job node J_j to a factory node F_f in Phase 1. As the scheduling process unfolds, the arcs in $\mathcal{E}$ and $\mathcal{K}$ are visually distinguished as dashed lines for unscheduled operations and solid lines for scheduled ones. Notably, job nodes J_j are not explicitly represented in $\mathcal{G}^H$. This design choice stems from the constraint that, once the initial operation of a job is assigned to a specific factory in Phase 1, all subsequent operations must be allocated to the same factory.

MDP--Lower-level Agent

Action

Phase 1

job-factory pairs (J, F)

Phase 2

operation-machine pairs (O, M)

Reward

4) Reward: We define the reward $r(s_t, a_t, s_{t+1})$ in the GDRL based on the weight quality difference between the solutions corresponding to states s_t and s_{t+1}, as shown in

$$r(s_t, a_t, s_{t+1}) = \frac{w(C_{\max}(s_t) - C_{\max}(s_{t+1}))}{C_{\max}(s_0)} + \frac{(1-w)(TEC(s_t) - TEC(s_{t+1}))}{\mathrm{TEC}(s_0)}. \quad (38)$$

Heterogeneous Graph Neural Network

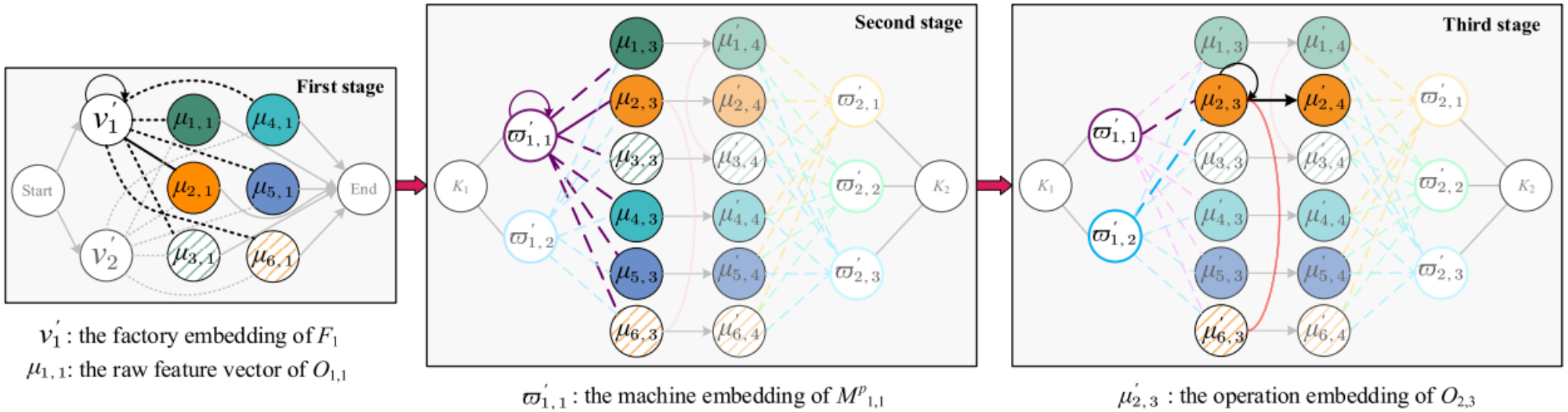


Fig. 7. Three-stage embedding scheme. Update of factory embedding v'_1, machine embedding $\varpi'_{1,1}$, and operation embedding $\mu'_{2,3}$ are highlighted for illustration.

Heterogeneous Graph Neural Network

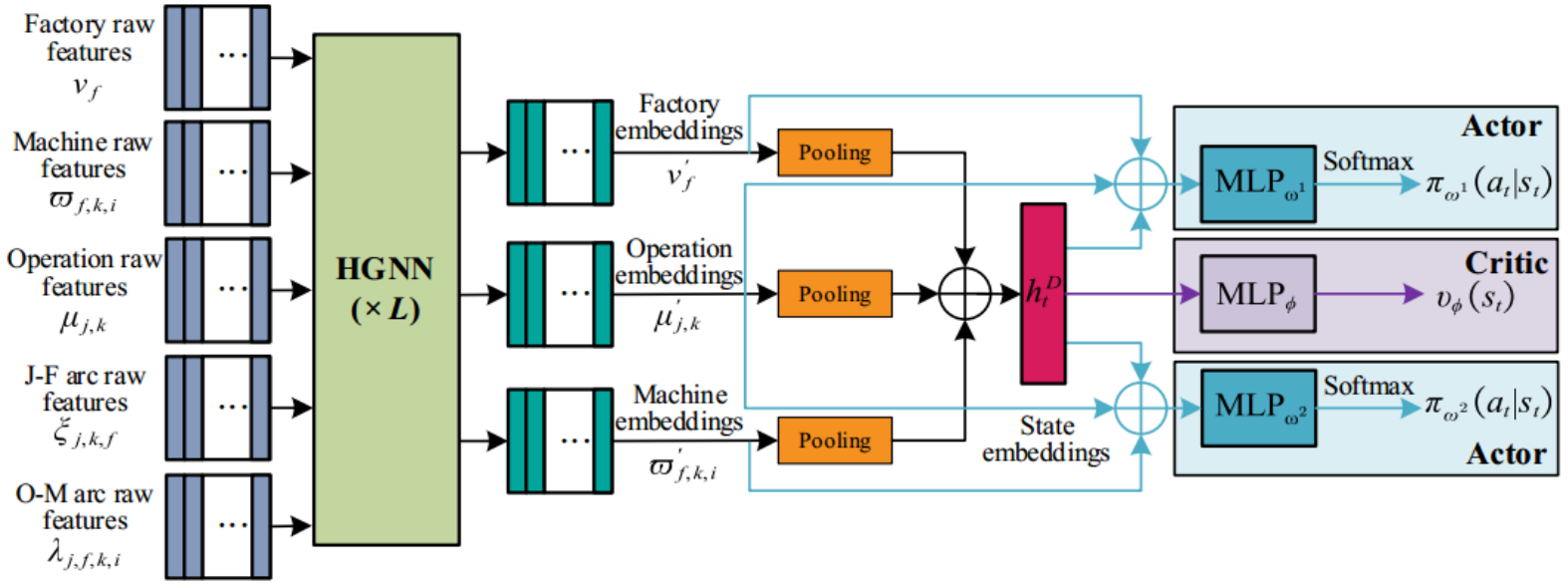


Fig. 4. The network architecture.

Factory node embedding	Machine node embedding	Operation node embedding
MLP	GAT	MLP

Contents

Experiment Results--Test instances and Baseline

1) Training and Testing Instances: The GDRL is trained and evaluated on synthetic instances of varying scales. The instance generation process follows a methodology similar to that employed in the referenced study. The parameters used for generating these instances are specified as follows, $n \in \{15, 20, 40, 50\}$, $\delta \in \{2, 3, 4, 5\}$, $m \in \{5, 10\}$, $s \in \{5, 10\}$, $r_k \sim U[2, 5]$, $p^1_{j,i} \sim U[10, 30]$, $p^2_{j,k} \sim U[10, 50]$, $\mathrm{st}^1_{j,j',i} \sim U[10, 20]$, $\mathrm{st}^2_{j,j',l,k} \sim U[10, 20]$, $tp_{j,f} \sim U[2, 5]$, $\beta^1_i \sim U[4, 8]$, $\beta^1_{k,l} \sim U[5, 10]$, $v_{k,l} \in \{1, 1.1, 1.2, 1.3, 1.4\}$, $\gamma = 1$, $\theta = 0.5$, $\mu = 3$, $|JM| = 0.75n$, and $|JS| = 0.25n$, where U denotes the uniform distribution, and all other symbols are thoroughly defined in Section III-A. Notably, 64 groups of instances are generated, covering all combinations of jobs, factories, machines, and stages. Each group consists 40 individual instances.

3) Baselines: As the ECDCSP-M problem remains largely unexplored, as discussed in Section II, we have selected a diverse set of baseline algorithms for comparison, encompassing DRL-based and RL-based approaches, metaheuristics, and multiobjective optimization methods:

1) *DRL-Based Method:* DRL-FJSP [21].
2) *RL-Based Method:* population-based iterative greedy algorithm (PBIGA) [47].
3) *Metaheuristic Algorithms:* hybrid scheduling algorithm (HAS) [19], multineighborhood-based multiobjective memetic algorithm (MMMA) [10].
4) *Multiobjective Optimization Method:* NSGA-II [18].

Experiment Results--Evaluation of Instances of Training Size

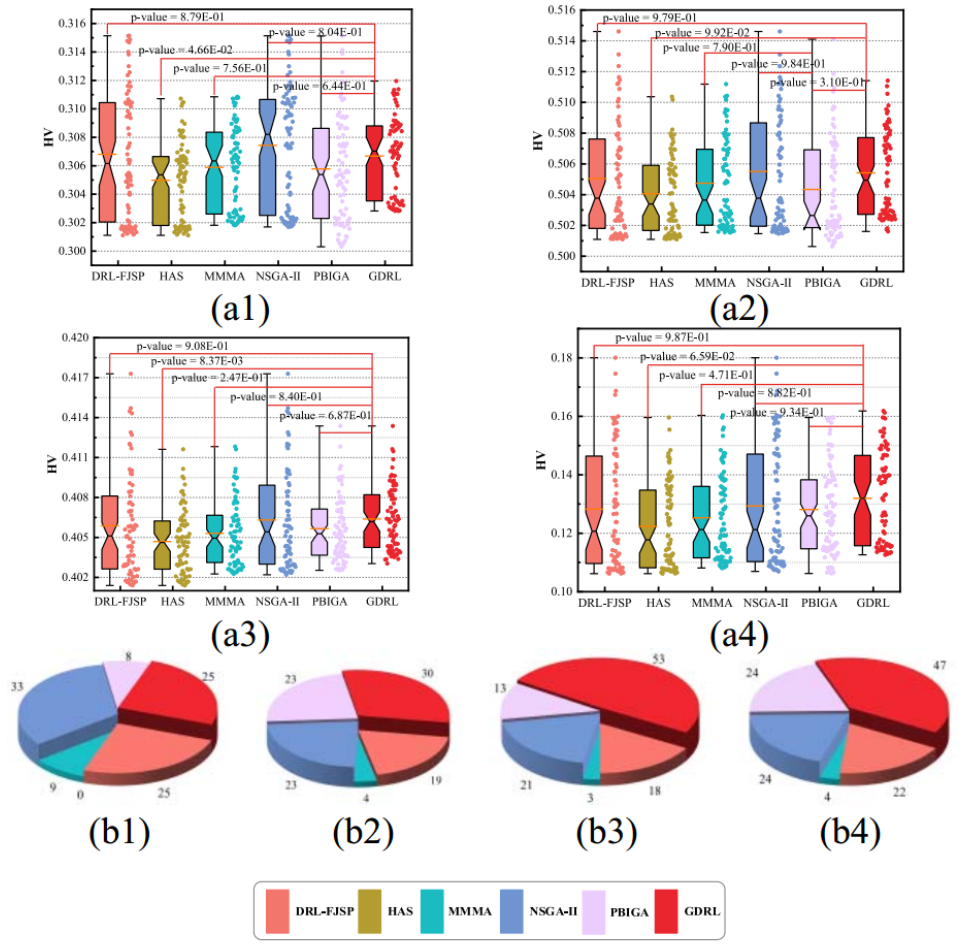
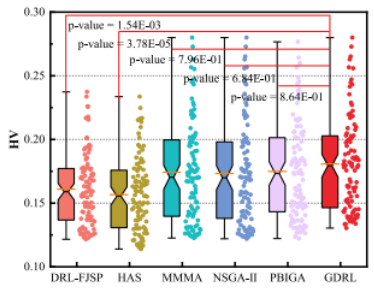


Fig. 9. Result of training size and peer comparison, (a) Boxplots of HV; (b) Win rate %. (a.1) $n = 15, m = 5, s = 5$ (a.2) $n = 15, m = 5, s = 10$ (a.3) $n = 15, m = 10, s = 5$ (a.4) $n = 15, m = 10, s = 10$ (b.1) $n = 15, m = 5, s = 5$ (b.2) $n = 15, m = 5, s = 10$ (b.3) $n = 15, m = 10, s = 5$ (b.4) $= 15, m = 10, s = 10$.

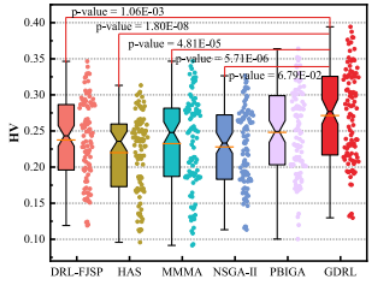
Experiment Results--Generalization Performance on Large-Sized Instances



(a1)

Algorithms	Subsets $\alpha = 0.05$		
	1	2	3
HAS	0.1566	/	/
DRL-FJSP	0.1610	0.1610	/
NSGA-II	/	0.1733	0.1733
MMMA	/	0.1742	0.1742
PBIGA	/	0.1749	0.1749
GDRL	/	/	0.1807
p-value	9.57E-01	6.72E-02	6.84E-01

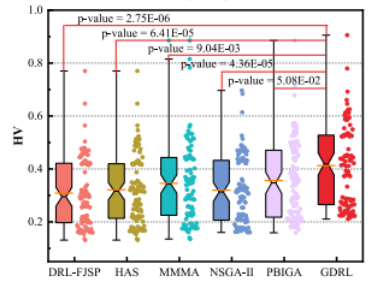
(b1)



(a2)

Algorithms	Subsets $\alpha = 0.05$		
	1	2	3
HAS	0.2190	/	/
NSGA-II	0.2279	0.2279	/
MMMA	0.2316	0.2316	/
DRL-FJSP	0.2376	0.2376	/
PBIGA	/	0.2481	0.2481
GDRL	/	/	0.2714
p-value	2.43E-01	1.62E-01	6.79E-02

(b2)



(a3)

Algorithms	Subsets $\alpha = 0.05$	
	1	2
DRL-FJSP	0.3089	/
NSGA-II	0.3199	/
HAS	0.3215	/
MMMA	0.3458	/
PBIGA	0.3565	0.3565
GDRL	/	0.4126
p-value	1.50E-01	5.08E-02

(b3)

Fig. 10. Results on large-sized instances and 95% Tukey HSD intervals in the ANOVA, (a) Boxplots of HV; (b) Homogeneous subsets of Tukey HSD (a.1) $n = 20$. (a.2) $n = 40$. (a.3) $n = 50$. (b.1) $n = 20$. (b.2) $n = 40$. (b.3) $n = 50$.

Experiment Results--Generalization Performance on Large-Sized Instances

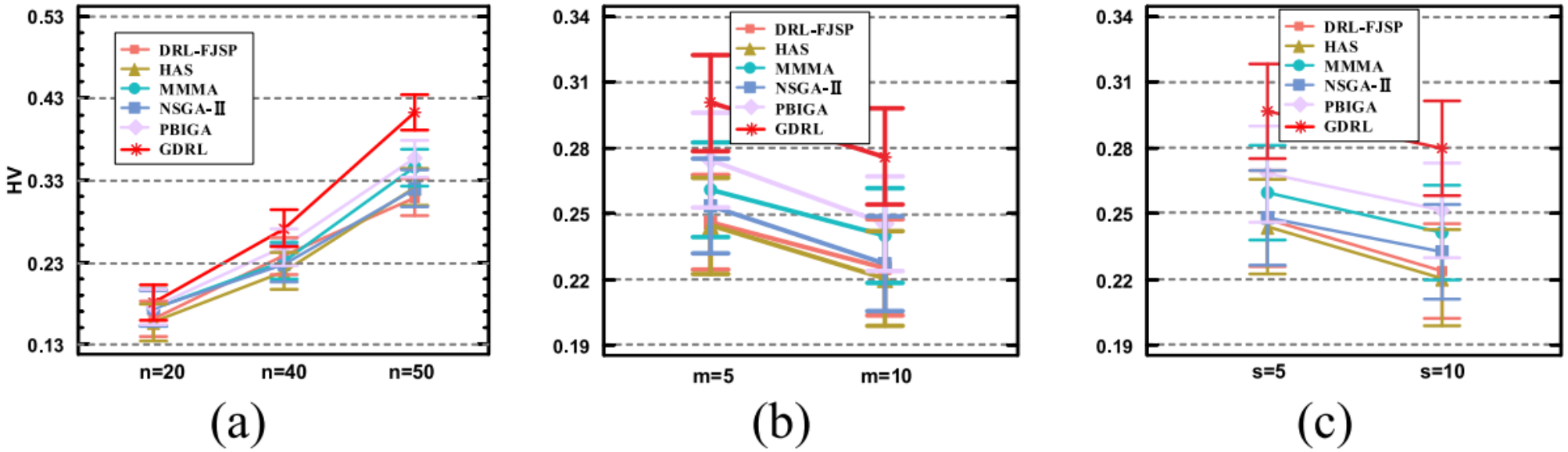


Fig. 11. Interactions and 95% Tukey HSD intervals in ANOVA. (a) Job. (b) Machine in Phase 1. (c) Stage in Phase 2.

Contents

Future Work

Several areas merit further investigation. First, real-world production scheduling often involves more complexities beyond mating operations, such as uncertainties like machine failures and urgent order insertions. Incorporating these elements could enhance the robustness and adaptability of the scheduling model. Additionally, the end-to-end DRL approach could benefit from further improvements in feature extraction of the scheduling state, as well as the integration of advanced search mechanisms, to boost training efficiency and overall performance.

Inspiration

Possible directions for future work: heterogeneous scale, with AGV (automated guided vehicle) integration, dynamic scheduling, real-time scheduling.

Extension of the current work: Distributed multi-factory scheduling based on GNN.

Thank you for listening