TAEPPO: Topology-Aware Relational Attention Networks with Adaptive-Entropy Proximal Policy Optimization for Flexible Job Shop Scheduling

Anonymous Authors

Abstract

The Flexible Job-Shop Scheduling Problem (FJSP) is a cornerstone optimization challenge in smart manufacturing, requiring the simultaneous assignment of precedence-constrained operations to eligible machines. Given its NP-hard complexity, deep reinforcement learning (DRL) has emerged as a potent paradigm for deriving real-time scheduling policies with advanced scalability and robust generalization. However, existing DRL-based scheduling methods often face two critical limitations: the insufficient extraction of multi-granular topological relations and the inherent instability of algorithmic adaptation. To overcome these, we propose TAEPPO, a novel framework that integrates Topology-Aware Relational Attention Networks (TARAN) with Adaptive-Entropy Proximal Policy Optimization (AEPPO). Specifically, TARAN decomposes the scheduling graph into dual relational attention networks: one encodes operation embeddings by synthesizing local operation-level adjacency with global job-level context, while the other characterizes machine embeddings through heterogeneous operation-machine relations. Furthermore, AEPPO stabilizes the training trajectory via automated entropy regularization, dynamically balancing the exploration-exploitation trade-off across discrete decision-making spaces. Extensive experiments demonstrate that TAEPPO consistently outperforms both traditional PDRs and state-of-the-art DRL-based methods on diverse benchmarks, significantly improving feature representation extraction and policy optimization robustness.

1 Introduction

In the drive toward Industry 4.0 technologies [Xu *et al.*, 2021; Ahmed *et al.*, 2022], optimizing resource allocation and production sequencing is a paramount challenge to achieve operational excellence in modern manufacturing systems [Ahsan and Siddique, 2022; Mahmoodi *et al.*, 2024]. Within this domain, the Flexible Job-Shop Scheduling Problem (FJSP) is recognized as a central and critical NP-hard combinatorial optimization problem [Xie *et al.*, 2019; Dauzère-Pérès *et al.*, 2024], defined by the management of a finite set of jobs and machines. Each job consists of an ordered collection of operations governed by inherent precedence constraints, with each operation needing to be assigned to one of several compatible machines [Lei *et al.*, 2022]. The objective of this complex problem is typically to maximize throughput efficiency by minimizing the makespan, which refers to the maximum completion time [Chaudhry and Khan, 2016].

Recently, deep reinforcement learning (DRL) has been widely adopted to address FJSP [Han and Yang, 2021; Liu *et al.*, 2022; Yuan *et al.*, 2024; Wang *et al.*, 2025], outperforming traditional exact methods [Meng *et al.*, 2020], metaheuristics [Gao *et al.*, 2019], and heuristic priority dispatch rules (PDRs) [Demir and Yilmaz, 2021] in terms of both computational efficiency and scheduling quality. DRL-based scheduling methodologies model the decision-making process as a Markov Decision Process (MDP) [Lei *et al.*, 2023], providing an end-to-end learning framework [Wang *et al.*, 2021] to define states, actions, state transitions and rewards. This formulation enables agents to autonomously learn and optimize dynamic PDRs through iterative environment interactions.

The intricate nature of FJSP, compounded by stringent intra-job precedence constraints and machine assignment flexibility, demands robust structural representations to navigate the complex solution space [Smit *et al.*, 2025]. Consequently, there is a growing trend towards integrating specialized graph neural networks (GNNs) [Zhou *et al.*, 2020; Corso *et al.*, 2024] into DRL architectures to encode high-dimensional correlations and dynamic dependencies within the scheduling state. Specifically, recent studies have effectively utilized the disjunctive graph [Brandimarte, 1993; Zhang *et al.*, 2024] for comprehensive state representation, achieving remarkable advancements within this field [Song *et al.*, 2022; Wang *et al.*, 2023; Zhao *et al.*, 2025].

However, existing DRL-based scheduling methods exhibit two critical challenges. First, they typically treat the scheduling graph as a monolithic entity, thereby impeding the distillation of inherent multi-granular topological relations. These models predominantly restrict operation embeddings to local precedence chains [Song *et al.*, 2022; Wang *et al.*, 2023], focusing exclusively on immediate adjacency relations (predecessors and successors) while overlooking the global job-level context. However, such global semantics are critical to evaluate both scheduled and remaining workloads, enabling the agent to optimize resource allocation across the entire production horizon. Second, most prevailing methods directly employ vanilla DRL algorithms [Yuan *et al.*, 2024; Zhao *et al.*, 2025], such as standard proximal policy optimization (PPO) [Schulman *et al.*, 2017] and deep Q-learning (DQN) [Du *et al.*, 2022], without domain-specific adaptation. These generic implementations are often ill-suited for navigating the dynamic, discrete decision-making spaces and the intricate resource bottlenecks, resulting in limited robustness and suboptimal performance.

To address the aforementioned limitations, we present an advanced framework, named TAEPPO, which synergistically integrates topology-aware relational attention networks (TARAN) with adaptive-entropy proximal policy optimization (AEPPO). Firstly, the TARAN module is engineered to extract multi-granular structural relations by decomposing the scheduling graph into dual relational attention streams. By incorporating local operation-level adjacency and global job-level context for operation embeddings, and mapping heterogeneous operation-machine relations for machine embeddings, TARAN facilitates a multi-granular feature extraction that transcends conventional precedence chains. Secondly, AEPPO stabilizes learning trajectories through an adaptive entropy regularization to dynamically balance stochastic exploration and deterministic exploitation, thereby accelerating convergence robustness and improving policy generalization relative to the vanilla PPO algorithm. The main contributions are stated as follows:

- We propose the TARAN module, which decouples graph feature extraction to characterize operation embeddings by incorporating local operation-level adjacency and global job-level context relations, while simultaneously deriving machine embeddings by encoding heterogeneous operation-machine relations.
- We develop the AEPPO algorithm, which leverages an adaptive entropy loss to mitigate training volatility, ensuring stabilized learning trajectories and enhanced policy robustness compared to the standard PPO.
- Integrating the aforementioned components, extensive experiments demonstrate that TAEPPO exhibits superior performance on multiple benchmarks.

2 Related Work

2.1 Traditional Methods

Historically, FJSP research has evolved through three paradigms. Initially, exact methods, such as mathematical and constraint programming [Demir and İşleyen, 2013; Yao *et al.*, 2024] solved FJSP by formulating rigorous mixed-integer linear programming (MILP) models. Although these approaches guaranteed optimality through systematic exploration, their exponential complexity renders them impractical for large-scale applications. To mitigate this, metaheuristics were developed, such as evolutionary strategies [Pan *et al.*, 2022; Chen *et al.*, 2025] and swarm optimization [Nouiri *et al.*, 2018; Xu *et al.*, 2024; Shi *et al.*, 2023], employing stochastic search procedures to identify high-quality solutions. However, these methods still struggle with significant overhead in real-time scenarios. Consequently, heuristic priority dispatching rules (PDRs) [Chen and Matis, 2013; Demir and Yilmaz, 2021] are preferred for their efficiency, but often yield suboptimal solutions by relying on simplistic logic such as Most Operations Remaining (MOR) and Most Work Remaining (MWKR) [Sels *et al.*, 2012].

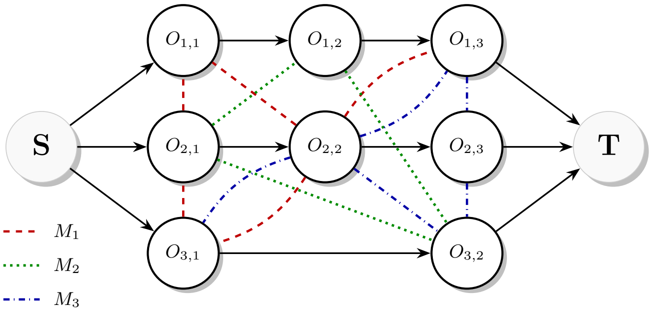

Figure 1: The disjunctive graph of an FJSP instance. The solid arcs represent the precedence constraints within each job, whereas the dashed arcs denote the disjunctive relationships among operations competing for the shared machines.

2.2 DRL-Based Methods

Recently, integrating DRL with GNNs has achieved significant success in addressing FJSP [Zhao *et al.*, 2023; Li *et al.*, 2025; Wu *et al.*, 2025]. The development of effective DRL-based scheduling frameworks has been driven by the introduction of various graph representations, which take advantage of the inherent compatibility of the disjunctive graph [Brandimarte, 1993], employing advanced GNNs to extract rich structural information [Huang *et al.*, 2025; Smit *et al.*, 2025]. In terms of specific methods, heterogeneous GNNs (HGNN) encode intricate relational patterns [Song *et al.*, 2022], while dual attention networks (DANIEL) meticulously capture dynamic shop-floor production features [Wang *et al.*, 2023]. Furthermore, lightweight MLPs (LM-DRL) streamline architectures for efficiency [Yuan *et al.*, 2024], while dual operation aggregation GNNs (DOAGNN) model complex dependencies across different jobs [Zhao *et al.*, 2025]. For decision-making, vanilla DRL algorithms, such as standardized proximal policy optimization (PPO) [Schulman *et al.*, 2017] and deep Q-learning (DQN) [Du *et al.*, 2022], are widely applied to facilitate robust optimization of scheduling models. Consequently, developing more sophisticated GNN and refining more robust DRL algorithms constitute a frontier of current investigation.

3 Problem Formulation

The Flexible Job-Shop Scheduling Problem extends the classical Job-Shop Scheduling Problem (JSP) by removing fixed machine constraints. Specifically, it involves n jobs $J = \{J_1, J_2, \ldots, J_n\}$ and m machines $M = \{M_1, M_2, \ldots, M_m\}$, where each job J_i consists of a sequence of n_i operations $O_i = \{O_{i,1}, O_{i,2}, \ldots, O_{i,n_i}\}$ to be processed in order. The set of all operations is denoted by $O = \bigcup_i O_i$. Unlike JSP, each operation $O_{i,j}$ is assignable to any machine in a compatible subset $M_{i,j} \subseteq M$, where the processing time $p_{i,j,k}$ varies depending on the selected machine M_k. The completion time of $O_{i,j}$ can be denoted as $C_{i,j}$, and the objective is to minimize the makespan $C_{\max}$ that represents the completion time of all operations:

$$C_{\max} = \max_{i,j}\{C_{i,j}\}. \tag{1}$$

The scheduling of the FJSP is governed by the following several constraints:

- **Precedence constraint**: operations within the same job follow a strictly predefined sequence.
- **Assignment constraint**: each operation be allocated to exactly one compatible machine.
- **Exclusivity constraint**: each individual machine can process at most one operation at a time without interruption or preemption.

We denote the disjunctive graph $G = (O', C, D)$ for FJSP, where O' includes all operations and dummy nodes $\{S, E\}$. Conjunctive arcs C enforce job sequences, while disjunctive arcs D represent compatible machine assignments. Upon assignment of an operation to a specific machine, the corresponding disjunctive arcs in D are transformed into directed edges, while all conflicting edges are simultaneously eliminated to maintain a feasible schedule. Figure 1 illustrates the disjunctive graph of an FJSP instance.

4 Methodology

In this section, we present the proposed TAEPPO in detail. We formulate FJSP as an Markov Decision Process [Song *et al.*, 2022; Wang *et al.*, 2023], where decision-making follows an iterative paradigm. At each step, we select a compatible operation-machine pair via the TARAN module that decouples feature extraction by fusing local operation-level adjacency and global job-level context relations for operation embeddings while encoding heterogeneous operation-machine relations for machine embeddings. These embeddings are then concatenated and fed into an actor-critic network refined via the AEPPO algorithm, which incorporates an adaptive entropy adjustment mechanism to balance exploration and exploitation, ensuring stabilized learning trajectories. The framework of TAEPPO is illustrated in Figure 2.

4.1 Markov Decision Process

We formulate the FJSP within a unified MDP framework that governs the entire scheduling trajectory, from the initial unscheduled state to a terminal state. The following sections present these MDP components, including state representation, action space, reward mechanism, and state transition.

State Representation

At each step t, the state of the system s_t consists of operations, machines, and their pair of operations-machines. For each operation $O_{i,j}$, machine M_k, and compatible pair $(O_{i,j}, M_k)$, we define their attributes as feature vectors $h_{O_{i,j}} \in \mathbb{R}^6$, $h_{M_k} \in \mathbb{R}^4$, and $h_{(O_{i,j},M_k)} \in \mathbb{R}^4$, respectively. These vectors are engineered to encapsulate critical real-time status indicators and inherent processing capabilities, thereby providing the model with a granular and comprehensive view of the dynamic scheduling environment. Details of these feature vectors are documented in **Appendix A**.

Action Space

Let $A(t)$ denote the action space at step t, with $a_t \in A(t)$ representing a specific action. To satisfy precedence constraints, the operation sequencing is restricted to the immediate successor of the most recently scheduled one within each job. Consequently, the candidate operations are upper-bounded by the job count, yielding a maximum action space cardinality of $n \times m$ across all machine-assignment permutations.

Reward Mechanism

The completion time of operation $O_{i,j}$ is defined as its actual finish time if already scheduled; otherwise, it is iteratively estimated based on the finish time of its predecessor $O_{i,j-1}$:

$$C(O_{i,j}) = C(O_{i,j-1}) + \frac{\sum_{M_k \in M_{i,j}} p_{i,j,k}}{|M_{i,j}|}, \tag{2}$$

where $M_{i,j} \subseteq M$ denotes the set of eligible machines for $O_{i,j}$. The reward r_t guides the policy towards minimizing the makespan $C_{\max}(s_t)$, defined as the incremental reduction in estimated makespan. With a discount factor $\gamma = 1$, the cumulative reward over $T = |O|$ steps is formulated as follows:

$$\begin{aligned} \sum_{t=0}^{T-1} r_t &= \sum_{t=0}^{T-1} [C_{\max}(s_t) - C_{\max}(s_{t+1})] \\ &= C_{\max}(s_0) - C_{\max}(s_T) \xrightarrow{C_{\max}(s_0)=0} -C_{\max}(s_T). \end{aligned} \tag{3}$$

Given that $C_{\max}(s_0) = 0$ for an empty initial schedule, the cumulative reward yields a direct and intrinsic correspondence between the maximization of the reward and the minimization of the final makespan.

State Transition

Upon taking an action a_t, the environment undergoes a controlled transition from the current state s_t to the subsequent state s_{t+1}, updating the scheduling configuration and reflecting the real-time allocation of resources within the system.

4.2 TARAN Module

Specifically, the proposed Topology-Aware Relational Attention Networks (TARAN) bifurcates the scheduling graph learning into dual relational attention streams: one synchronizes local operation-level adjacency with global job-level context relations to generate enriched operation embeddings, while the other maps heterogeneous operation-machine relations into machine embeddings. By integrating these multigranular features, TARAN enables a comprehensive state encoding that discerns both fine-grained task requirements and systemic resource constraints.

Operation Embeddings

Characterizing intricate relations among operations is vital in FJSP. Local operation-level adjacency delineates precedence constraints, while global job-level context captures workload requirements. Integrating these relations enables the model to bridge sequential logic with global workload distributions, which is essential for optimizing scheduling objectives.

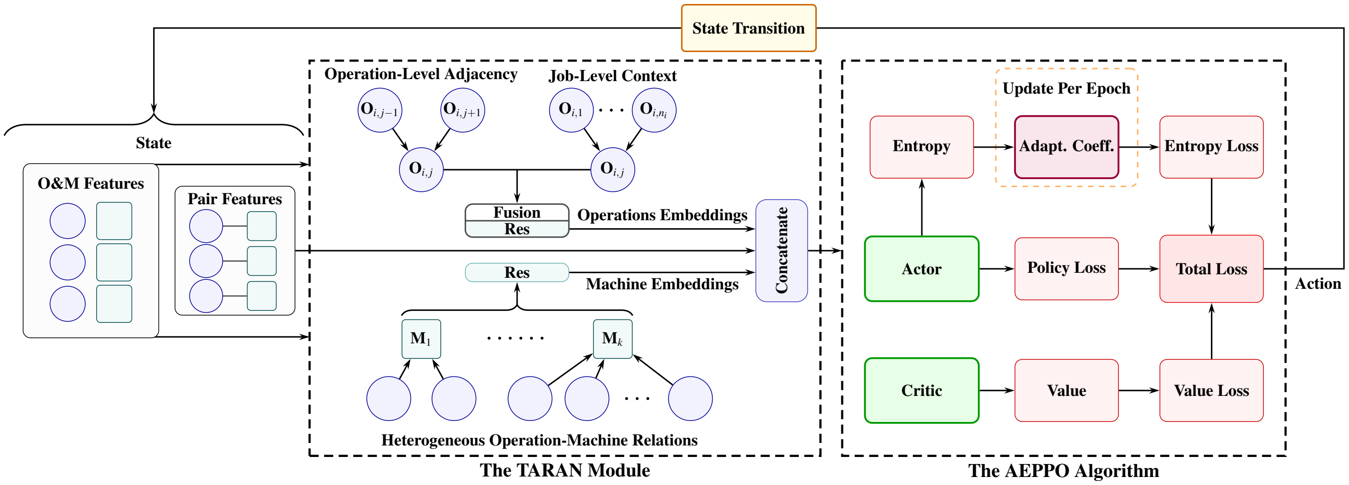

Figure 2: The overall framework of TAEPPO. The TARAN module fuses operation-job features for operation embeddings and encodes operation-machine relations for machine embeddings, while the AEPPO algorithm employs adaptive entropy to stabilize learning trajectories.

For each operation $O_{i,j}$ with raw features $h_{O_{i,j}} \in \mathbb{R}^6$, the attention coefficient between operations $O_{i,j}$ and $O_{i,u}$ in the same job is computed via a shared attention mechanism:

$$e_{i,j,u} = \text{LeakyReLU}(\mathbf{a}_{src}^{\mathrm{T}} \mathbf{W} h_{O_{i,j}} + \mathbf{a}_{dst}^{\mathrm{T}} \mathbf{W} h_{O_{i,u}}), \quad (4)$$

where $\mathbf{a}_{src}$ and $\mathbf{a}_{dst} \in \mathbb{R}^{d_{out}}$ denote the learnable attention vectors, while $\mathbf{W} \in \mathbb{R}^{d_{out} \times 6}$ projects the input features into the latent space. The module bifurcates attention aggregation to capture operation-level adjacency and job-level context:

Operation-Level Adjacency Relations: We denote the attention weight $\alpha_{i,j,u}^{adj}$ to measure the correlation between $O_{i,j}$ and its neighbor $O_{i,u}$. This weight is obtained by normalizing the raw attention coefficients $e_{i,j,u}$:

$$\alpha_{i,j,u}^{adj} = \frac{\exp(e_{i,j,u})}{\sum_{v \in N_{i,j}^{adj}} \exp(e_{i,j,v})}, \quad (5)$$

where $N_{i,j}^{adj} = \{O_{i,j-1}, O_{i,j}, O_{i,j+1}\}$ denotes topologically adjacent operations. The operation-level features for $O_{i,j}$ are synthesized through a weighted aggregation:

$$h_{O_{i,j}}^{adj} = \sum_{u \in N_{i,j}^{adj}} \alpha_{i,j,u}^{adj} \mathbf{W} h_{O_{i,u}}. \quad (6)$$

As initial and terminal operations lack predecessors or successors, we employ an adjacency matrix masking mechanism to filter non-existent dependencies.

Job-Level Context Relations: To capture the context of job J_i, the attention weight $\alpha_{i,j,x}^{job}$ is denoted to quantify the relative importance between $O_{i,j}$ and $O_{i,x}$ within the same job, derived by normalizing the attention coefficients $e_{i,j,x}$:

$$\alpha_{i,j,x}^{job} = \frac{\exp(e_{i,j,x})}{\sum_{y \in N_{i,j}^{job}} \exp(e_{i,j,y})}, \quad (7)$$

where $N_{i,j}^{job}$ denotes all operations belonging to the same job. The aggregated job-level context for $O_{i,j}$ is calculated as:

$$h_{O_{i,j}}^{job} = \sum_{x \in N_{i,j}^{job}} \alpha_{i,j,x}^{job} \mathbf{W} h_{O_{i,x}}. \quad (8)$$

By constructing an intra-job matrix, we can employ a masking mechanism to filter out operations from different jobs.

Channel-Wise Fusion: To balance operation-level adjacency and job-level context, we employ a learnable fusion mechanism. The final embedding $h_{O_{i,j}}^{emb}$ aggregates these representations using channel-wise attention weights and a residual connection to ensure training stability:

$$\begin{aligned} [\lambda_z^{adj}, \lambda_z^{job}] &= \text{Softmax}(\mathbf{\Lambda}_z), \\ h_{O_{i,j}}^{emb} &= \lambda_z^{adj} \odot h_{O_{i,j}}^{adj} + \lambda_z^{job} \odot h_{O_{i,j}}^{job} + h_{O_{i,j}}^{res}, \end{aligned} \quad (9)$$

where $\odot$ denotes the element-wise product and $\mathbf{\Lambda} \in \mathbb{R}^{d_{out} \times 2}$ is the learnable weight matrix. This fusion enables the model to dynamically prioritize critical features for each operation.

Machine Embeddings

To capture complex operation-machine relations, we propose a heterogeneous networks that maps task-relevant operation features into machine representations. For each machine M_k with raw input h_{M_k}, we denote its neighboring operations as a set N_k. The attention coefficient between M_k and its neighboring operations $O_{i,j} \in N_k$ is defined as follows:

$$e_{i,j,k} = \text{LeakyRelu}(\mathbf{a}_o^{\mathrm{T}} \mathbf{W}_o h_{O_{i,j}} + \mathbf{a}_m^{\mathrm{T}} \mathbf{W}_m h_{M_k})), \quad (10)$$

where $\mathbf{W}_o$ and $\mathbf{W}_m$ represent learnable weight matrices that project features into a shared latent space, while $\mathbf{a}_o$ and $\mathbf{a}_m$ denote the corresponding weight vectors for operations and machines, respectively.

Operation-Machine Relations: Similarly, these coefficients are then normalized via a softmax function across all neighboring operations:

$$\alpha_{i,j,k} = \frac{\exp(e_{i,j,k})}{\sum_{O_{u,v} \in N_k} \exp(e_{u,v,k})}. \quad (11)$$

By incorporating a residual connection, the final embedding of machine M_k is obtained as follows:

$$h_{M_k}^{emb} = \sum_{O_{i,j} \in N_k} \alpha_{i,j,k} \mathbf{W}_o h_{O_{i,j}} + h_{M_k}^{res}. \quad (12)$$

Through this formulation, the model effectively encapsulates the operation-machine relations, enabling the machine embeddings to assimilate heterogeneous structural information.

State Embeddings

The state embedding is synthesized by performing and concatenating average pooling over both operation embeddings and machine embeddings as follows:

$$h_{s_t} = \left[\frac{1}{|O|} \sum_{O_{i,j}\in O} h^{emb}_{O_{i,j}} \middle\| \frac{1}{|M|} \sum_{M_k \in M} h^{emb}_{M_k} \right]. \tag{13}$$

This state embedding integrates operation-level and machine-level features, delivering a comprehensive overview of the global state for each decision-making step.

4.3 AEPPO Algorithm

To solve FJSP, we employ Adaptive-Entropy Proximal Policy Optimization (AEPPO), enhanced with a dynamic entropy regularization mechanism. This ensures a robust balance between exploration and exploitation in the high-dimensional discrete action space of scheduling. Diverging from standard PPO, AEPPO utilizes stochastic mini-batch sampling from collected trajectories to stabilize gradient updates and improve data efficiency. The algorithmic procedure is detailed in Algorithm 1.

Actor-Critic

The policy and value functions are implemented through an actor-critic architecture, where two distinct multilayer perceptrons (MLPs) serve as the actor π_θ and the critic v_ϕ, respectively. For each action $a_t = (O_{ij}, M_k) \in A_t$, the actor computes priority index $P(a_t|s_t)$ by processing concatenated operation, machine embeddings, and pair features, and the action selection probability $\pi_\theta(a_t|s_t)$ is normalized over $A(t)$:

$$\begin{aligned} P(a_t|s_t) &= \mathrm{MLP}_\theta \left[h^{emb}_{O_{i,j}} \| h^{emb}_{M_k} \| h_{(O_{i,j},M_k)} \right], \\ \pi_\theta(a_t|s_t) &= \frac{\exp(P(a_t|s_t))}{\sum_{a'_t \in A(t)} \exp(P(a'_t|s_t))}. \end{aligned} \tag{14}$$

Furthermore, the critic estimates the state value $v_\phi(s_t)$ using the state embedding:

$$v_\phi(s_t) = \mathrm{MLP}_\phi(h_{s_t}). \tag{15}$$

Policy and Value Loss

To ensure stable updates, the PPO-clip objective employs the ratio $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$ that denotes the probability ratio between current and old behavioral policies to penalize policy deviations outside the $[1-\epsilon, 1+\epsilon]$ interval:

$$\begin{aligned} \mathcal{L}^{POL} = C_p \cdot \{ -\mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \\ \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \}, \end{aligned} \tag{16}$$

where ϵ is the clipping hyperparameter, $\hat{A}_t$ is the advantage estimation, and C_p is the policy loss coefficient.

The value loss is the Mean Squared Error (MSE) between the predicted value and discounted rewards R_t:

$$\mathcal{L}^{VAL} = C_v \cdot \mathbb{E}_t \left[(v_\phi(s_t) - R_t)^2 \right], \tag{17}$$

where C_v denotes the value loss coefficient and $v_\phi(s_t)$ represents the estimated state value.

Algorithm 1 The training process of AEPPO

Input: Pre-sampled training data D_{tra}, validation data D_{val}
Parameter: Initial policy parameters π_θ, value parameters v_ϕ, entropy coefficient parameters C_e
Output: The trained optimal π_θ

```
for iteration $= 1, 2, \ldots, T$ do
    Collect trajectories $\mathcal{T} = \{s_t, a_t, r_t, s_{t+1}\}$ of $D_{tra}$.
    Estimate target entropy $H_{tar}$ using Eq. (18), compute estimated advantages $\hat{A}_t$ and discounted rewards $R_t$.
    for epoch $= 1, \ldots, K$ do
        for each minibatch $B_{min} \sim D_{tra}$ do
            Calculate losses $\{\mathcal{L}^{POL}, \mathcal{L}^{VAL}, \mathcal{L}^{ENT}\}$ according to Eqs. (16), (17), and (20).
            Calculate total loss $\mathcal{L}^{TOT}$ using Eq. (21).
            Update parameters $\pi_\theta, v_\phi$.
        end for
        Update entropy coefficient $C_e$ using Eq. (19).
    end for
    Resample $|D_{tra|}$ FJSP instances every $T_{tra}$ iteration.
    Validate $\pi_\theta$ on $D_{val}$ every $T_{val}$ iteration.
end for
return Optimal $\pi_\theta$
```

Adaptive Entropy Loss

In FJSP, the feasible action space cardinality varies across decision-making stages. We implement an automated tuning loop for entropy coefficient C_e, defining target entropy H_{tar} based on the average number of eligible actions A_e:

$$\begin{aligned} A_e &= \sum_{a_t \in A(t)} m(a_t|s_t), \\ H_{tar} &= \beta \cdot \ln(A_e + 10^{-8}), \end{aligned} \tag{18}$$

where $m(a_t|s_t) \in \{0, 1\}$ is a binary indicator that denotes the feasibility of action a_t in state and β is a conservative empirical target density. While the policy updates for K epochs per iteration, $\log C_e$ is optimized epoch-wise by minimizing:

$$\mathcal{L}(C_e) = \log C_e \cdot (H_{avg} - H_{tar}), \tag{19}$$

where H_{avg} is the mean entropy observed during the epoch. This mechanism ensures that C_e increases when the policy becomes overly deterministic and decreases once the target exploration level is reached. The entropy loss is denoted as:

$$\mathcal{L}^{ENT} = -C_e \cdot \hat{\mathbb{E}}_t \left[H_s(\pi_\theta(\cdot|s_t)) \right], \tag{20}$$

where H_s denotes the Shannon entropy of the action distribution. Our AEPPO diverges from Soft Actor-Critic [Haarnoja *et al.*, 2018] by employing a dynamic entropy target tailored to varying action spaces with specialized update formulation. Theoretical justification is provided in **Appendix B**.

Total Loss

Combining the policy loss, the value loss, and the adaptive entropy loss, the total loss is defined as follows:

$$\mathcal{L}^{TOT} = \mathcal{L}^{POL} + \mathcal{L}^{VAL} + \mathcal{L}^{ENT}. \tag{21}$$

In summary, this composite loss function empowers TAEPPO with enhanced numerical stability and robust convergence, ensuring reliable performance when navigating the complex combinatorial decision-making space of FJSP environments.

Methods		Brandimarte		Rdata		Edata		Vdata	
		$C_{max}\downarrow$	$Gap\downarrow$	$C_{max}\downarrow$	$Gap\downarrow$	$C_{max}\downarrow$	$Gap\downarrow$	$C_{max}\downarrow$	$Gap\downarrow$
Traditional PDRs	FIFO	205.56	31.82%	1087.12	17.25%	1244.92	20.83%	982.89	7.58%
	SPT	237.52	44.88%	1200.41	29.47%	1312.84	26.79%	1082.88	18.20%
	MOR	200.36	28.08%	1066.73	15.07%	1227.07	19.24%	966.01	5.68%
	MWKR	201.74	28.91%	1053.10	13.86%	1219.01	18.60%	952.00	4.22%
HGNN	10×5	201.00	27.83%	1030.83	11.15%	1187.47	15.53%	955.90	4.25%
	15×10	197.70	25.39%	1031.33	11.14%	1182.08	15.00%	954.33	4.02%
DANIEL	10×5	185.70	13.58%	1031.63	11.42%	1194.98	16.33%	944.85	3.28%
	15×10	184.40	12.97%	1040.05	12.07%	1175.53	14.41%	948.73	3.75%
LMDRL	10×5	186.80	13.24%	1041.83	12.09%	1187.93	15.54%	963.50	5.37%
DOAGNN	10×5	199.60	30.43%	1040.45	12.05%	1189.40	15.67%	958.80	4.73%
TAEPPO (ours)	10×5	187.10	14.73%	**1017.60**	**9.43%**	**1169.78**	**13.76%**	947.03	3.26%
	15×10	**183.00**	**11.71%**	1023.25	10.27%	1179.78	14.65%	**943.43**	**3.14%**

Table 1: Overall performance of our TAEPPO and baselines on four benchmarks. Bold indicates the best performance.

5 Experiments

5.1 Experimental Setup

Datasets

We conduct our experiments on four public benchmarks comprising various FJSP instances [Behnke and Geiger, 2012]: Brandimarte dataset (Mk01-10) [Brandimarte, 1993] and three sets of Hurink datasets (Rdata, Edata, and Vdata, each set consists of La01-40) [Hurink *et al.*, 1994], consistent with previous studies [Song *et al.*, 2022; Wang *et al.*, 2023; Yuan *et al.*, 2024; Zhao *et al.*, 2025]. Specifically, models are trained on two problem sizes: 10×5 and 15×10, where training instances are generated following [Song *et al.*, 2022]. Subsequently, the optimal trained policy is saved and evaluated on benchmarks to assess the performance of TAEPPO on out-of-distribution instances.

Baselines

Initially, we evaluate our proposed approach against four standard heuristic PDRs, including FIFO (First In First Out), SPT (Shortest Processing Time), MOR (Most Operations Remaining), and MWKR (Most Work Remaining). Furthermore, our comparative analysis focuses on four state-of-the-art DRL-based methodologies: HGNN [Song *et al.*, 2022], DANIEL [Wang *et al.*, 2023], LMDRL [Yuan *et al.*, 2024], and DOAGNN [Zhao *et al.*, 2025]. As elaborated in the Related Work section, these methods leverage sophisticated graph representations and various GNNs to address intricate scheduling challenges.

Evaluation Metrics

In addition to the makespan metric C_{max}, we also employ the average relative gap to comprehensively evaluate performance. This metric quantifies the deviation between the achieved C_{max} and the best-known solutions C_{max}^{best} reported in [Behnke and Geiger, 2012], defined as:

$$Gap = (\frac{C_{max}}{C_{max}^{best}} - 1) \times 100\%. \tag{22}$$

Implementation Details

The total number of training iterations, parallel batch size, and validation interval are set to 1000, 20, and 10, respectively. The PPO algorithm utilizes the AdamW optimizer with a learning rate of 3×10^{-4}. For testing, we exclusively evaluate and compare results using a greedy decoding strategy for all benchmarks. All experiments were conducted on a server equipped with an Intel Xeon Silver 4310 CPU and an NVIDIA GeForce RTX 4090 GPU. More details can be found in **Appendix C**. Our source code is available in the **Supplementary Material**.

5.2 Results and Analysis

Table 1 presents the comparative performance of TAEPPO compared to traditional PDRs and DRL-based baselines on the benchmark datasets. In general, TAEPPO achieves state-of-the-art results, outperforming all baselines in terms of both makespan and relative gap and demonstrating its exceptional capability in handling complex scheduling scenarios.

Compared to traditional PDRs, our method exhibits a markedly superior performance margin. In particular, it reduces the relative gap by more than 15% on the Brandimarte dataset and by more than 4% on both Rdata and Edata benchmarks, relative to the most competitive PDRs. These empirical results underscore the intrinsic limitations of heuristic PDRs in navigating the high-dimensional solution space of FJSP environments. In contrast, TAEPPO successfully derives dynamic and precise dispatching rules tailored to sophisticated topological relations.

Furthermore, TAEPPO outperforms DRL baselines with a substantial and measurable reduction in both the average makespan and the relative gap. Specifically, the TAEPPO model trained on 10×5 scale achieves an average makespan of 1017.60 and a mean gap of 9.43% on the Rdata benchmark, while all baseline models exceed 1030 and 11%, respectively. Similarly, on the Brandimarte dataset, the 15×10 model yields solutions with a makespan of 183.00 and a gap of 11.71%, surpassing the best-performing baselines. We at-

Components		Brandimarte		Rdata		Edata		Vdata	
TARAN	AEPPO	$C_{max}\downarrow$	$Gap\downarrow$	$C_{max}\downarrow$	$Gap\downarrow$	$C_{max}\downarrow$	$Gap\downarrow$	$C_{max}\downarrow$	$Gap\downarrow$
✗	✗	191.10	15.97%	1049.65	12.94%	1195.73	16.47%	971.15	5.86%
✓	✗	185.60	13.73%	1039.58	11.80%	1191.73	16.03%	961.65	5.06%
✗	✓	188.70	14.99%	1044.90	12.18%	1198.60	16.65%	985.38	7.55%
✓	✓	**183.00**	**11.71%**	**1023.25**	**10.27%**	**1179.78**	**14.65%**	**943.43**	**3.14%**

Table 2: Ablation study of two core components of TAEPPO on public benchmarks.

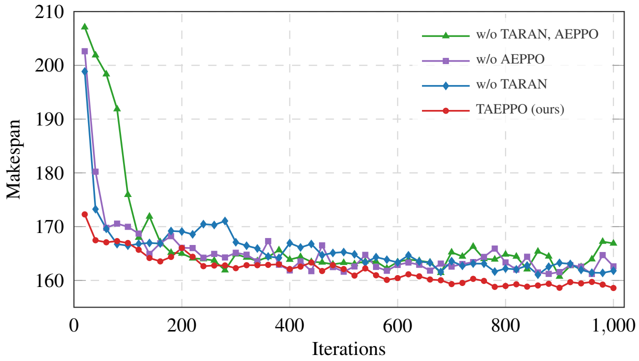

Figure 3: Training curves for the ablation study. Both TARAN and AEPPO modules significantly accelerate convergence while achieving optimal performance.

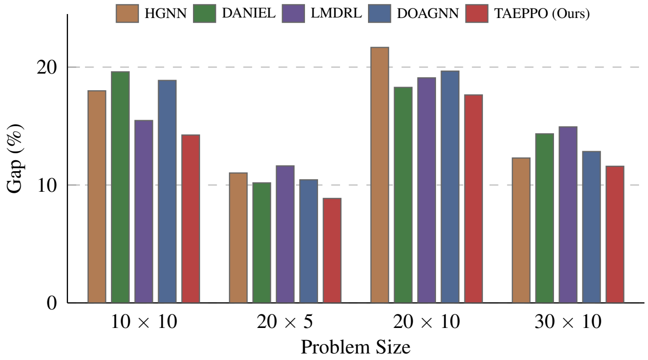

Figure 4: Generalization performance of TAEPPO compared to DRL-based baselines across unseen problem scales.

tribute this significant performance gain to the respective contributions of the proposed TARAN module and AEPPO algorithm. The former utilizes three topological relations to facilitate precise state encoding, while the latter enables the agent to converge to high-quality solutions with enhanced robustness via an adaptive entropy adjustment mechanism.

5.3 Ablation Study

Ablation studies are performed by substituting TARAN with parameter-equivalent MLPs and replacing AEPPO with vanilla PPO using a constant entropy coefficient to isolate their respective contributions. All variants are trained on 15×10 instances and evaluated on all benchmarks to quantify their relative and empirical effectiveness.

As illustrated by the training curves in Figure 3, variants lacking TARAN or AEPPO suffer from compromised optimization stability and inferior convergence rates, directly correlating with the observed degradation in generalization capability and final validation performance. Table 2 presents the ablation results of TAEPPO on public benchmarks. Without the TARAN module, the model fails to capture the intrinsic topological relations of the disjunctive graph, which limits its ability to encode comprehensive state representations. Similarly, in the absence of the AEPPO algorithm, the agent with fixed-entropy PPO struggles to manage exploration-exploitation trade-offs across varying problem scales, resulting in a significant drop in solution quality. Ultimately, the ablation study confirms that both TARAN and AEPPO are indispensable components, each contributing uniquely to the superior performance of TAEPPO.

5.4 Generalization Capability

A generalization experiment is conducted to rigorously evaluate the zero-shot performance of DRL-based methods trained on 10×5 instances when applied to out-of-distribution (OOD) problem scales. We report the average gap calculated across five independent instances for each problem scale within the Edata benchmark. As illustrated in Figure 4, TAEPPO consistently outperforms all baselines in four OOD scenarios. Consequently, the 10×5 TAEPPO model maintains its advantage in larger 20×10 and 30×10 cases, underscoring its superior generalization and robustness to unseen scales. This suggests TAEPPO captures scale-invariant topological relations rather than merely overfitting to the training distribution.

6 Conclusion

In this paper, we present TAEPPO, a novel DRL-based framework designed to tackle the inherent complexities of FJSP. The TAEPPO framework synergistically integrates TARAN and AEPPO, facilitating effective and adaptive scheduling by leveraging enhanced topological awareness and stabilized policy updates. Specifically, TARAN employs dual relations attention networks to capture the intricate structural relations between operations and machines, while AEPPO incorporates an automated entropy regularization scheme to dynamically balance exploration and exploitation throughout the solution space search. Experimental results across various benchmarks demonstrate that TAEPPO not only significantly outperforms both traditional PDRs and state-of-the-art DRL-based methods but also exhibits superior generalization capability when applied to unseen problem scales.

References

[Ahmed *et al.*, 2022] Imran Ahmed, Gwanggil Jeon, and Francesco Piccialli. From artificial intelligence to explainable artificial intelligence in industry 4.0: a survey on what, how, and where. *IEEE transactions on industrial informatics*, 18(8):5031–5042, 2022.

[Ahsan and Siddique, 2022] Md Manjurul Ahsan and Zahed Siddique. Industry 4.0 in healthcare: A systematic review. *International Journal of Information Management Data Insights*, 2(1):100079, 2022.

[Behnke and Geiger, 2012] Dennis Behnke and Martin Josef Geiger. Test instances for the flexible job shop scheduling problem with work centers. Arbeitspapier / research report, HELMUT-SCHMIDT-UNIVERSITÄT, 2012.

[Brandimarte, 1993] Paolo Brandimarte. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations research*, 41(3):157–183, 1993.

[Chaudhry and Khan, 2016] Imran Ali Chaudhry and Abid Ali Khan. A research survey: review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, 23(3):551–591, 2016.

[Chen and Matis, 2013] Binchao Chen and Timothy I Matis. A flexible dispatching rule for minimizing tardiness in job shop scheduling. *International Journal of Production Economics*, 141(1):360–365, 2013.

[Chen *et al.*, 2025] Xiaolong Chen, Junqing Li, Zunxun Wang, Qingda Chen, Kaizhou Gao, and Quanke Pan. Optimizing dynamic flexible job shop scheduling using an evolutionary multi-task optimization framework and genetic programming. *IEEE Transactions on Evolutionary Computation*, 2025.

[Corso *et al.*, 2024] Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. Graph neural networks. *Nature Reviews Methods Primers*, 4(1):17, 2024.

[Dauzère-Pérès *et al.*, 2024] Stéphane Dauzère-Pérès, Junwen Ding, Liji Shen, and Karim Tamssaouet. The flexible job shop scheduling problem: A review. *European Journal of Operational Research*, 314(2):409–432, 2024.

[Demir and İşleyen, 2013] Yunus Demir and S Kürşat İşleyen. Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling*, 37(3):977–988, 2013.

[Demir and Yilmaz, 2021] Yunus Demir and Hamid Yilmaz. An efficient priority rule for flexible job shop scheduling problem. *Journal of Engineering Research and Applied Science*, 10(2):1906–1918, 2021.

[Du *et al.*, 2022] Yu Du, Junqing Li, Chengdong Li, and Peiyong Duan. A reinforcement learning approach for flexible job shop scheduling problem with crane transportation and setup times. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5695–5709, 2022.

[Gao *et al.*, 2019] Kaizhou Gao, Zhiguang Cao, Le Zhang, Zhenghua Chen, Yuyan Han, and Quanke Pan. A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. *IEEE/CAA Journal of Automatica Sinica*, 6(4):904–916, 2019.

[Haarnoja *et al.*, 2018] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[Han and Yang, 2021] Baoan Han and J Yang. A deep reinforcement learning based solution for flexible job shop scheduling problem. *International Journal of Simulation Modelling*, 20(2):375–386, 2021.

[Huang *et al.*, 2025] Dailin Huang, Hong Zhao, Weiquan Tian, and Kangping Chen. A deep reinforcement learning method based on a multiexpert graph neural network for flexible job shop scheduling. *Computers & Industrial Engineering*, 200:110768, 2025.

[Hurink *et al.*, 1994] Johann Hurink, Bernd Jurisch, and Monika Thole. Tabu search for the job-shop scheduling problem with multi-purpose machines. *Operations-Research-Spektrum*, 15(4):205–215, 1994.

[Lei *et al.*, 2022] Kun Lei, Peng Guo, Wenchao Zhao, Yi Wang, Linmao Qian, Xiangyin Meng, and Liansheng Tang. A multi-action deep reinforcement learning framework for flexible job-shop scheduling problem. *Expert Systems with Applications*, 205:117796, 2022.

[Lei *et al.*, 2023] Kun Lei, Peng Guo, Yi Wang, Jian Zhang, Xiangyin Meng, and Linmao Qian. Large-scale dynamic scheduling for flexible job-shop with random arrivals of new jobs by hierarchical reinforcement learning. *IEEE Transactions on Industrial Informatics*, 20(1):1007–1018, 2023.

[Li *et al.*, 2025] Yuxin Li, Qingzheng Wang, Xinyu Li, Liang Gao, Ling Fu, Yanbin Yu, and Wei Zhou. Real-time scheduling for flexible job shop with agvs using multiagent reinforcement learning and efficient action decoding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2025.

[Liu *et al.*, 2022] Renke Liu, Rajesh Piplani, and Carlos Toro. Deep reinforcement learning for dynamic scheduling of a flexible job shop. *International Journal of Production Research*, 60(13):4049–4069, 2022.

[Mahmoodi *et al.*, 2024] Ehsan Mahmoodi, Masood Fathi, Madjid Tavana, Morteza Ghobakhloo, and Amos HC Ng. Data-driven simulation-based decision support system for resource allocation in industry 4.0 and smart manufacturing. *Journal of Manufacturing Systems*, 72:287–307, 2024.

[Meng *et al.*, 2020] Leilei Meng, Chaoyong Zhang, Yaping Ren, Biao Zhang, and Chang Lv. Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem. *Computers & industrial engineering*, 142:106347, 2020.

[Nouiri *et al.*, 2018] Maroua Nouiri, Abdelghani Bekrar, Abderezak Jemai, Smail Niar, and Ahmed Chiheb Ammari. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing*, 29(3):603–615, 2018.

[Pan *et al.*, 2022] Zixiao Pan, Ling Wang, Jie Zheng, Jing-Fang Chen, and Xing Wang. A learning-based multipopulation evolutionary optimization for flexible job shop scheduling problem with finite transportation resources. *IEEE Transactions on Evolutionary Computation*, 27(6):1590–1603, 2022.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Sels *et al.*, 2012] Veronique Sels, Nele Gheysen, and Mario Vanhoucke. A comparison of priority rules for the job shop scheduling problem under different flow time-and tardiness-related objective functions. *International Journal of Production Research*, 50(15):4255–4270, 2012.

[Shi *et al.*, 2023] Jiaxuan Shi, Mingzhou Chen, Yumin Ma, and Fei Qiao. A new boredom-aware dual-resource constrained flexible job shop scheduling problem using a two-stage multi-objective particle swarm optimization algorithm. *Information Sciences*, 643:119141, 2023.

[Smit *et al.*, 2025] Igor G Smit, Jianan Zhou, Robbert Reijnen, Yaoxin Wu, Jian Chen, Cong Zhang, Zaharah Bukhsh, Yingqian Zhang, and Wim Nuijten. Graph neural networks for job shop scheduling problems: A survey. *Computers & Operations Research*, 176:106914, 2025.

[Song *et al.*, 2022] Wen Song, Xinyang Chen, Qiqiang Li, and Zhiguang Cao. Flexible job-shop scheduling via graph neural network and deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, 19(2):1600–1610, 2022.

[Wang *et al.*, 2021] Libing Wang, Xin Hu, Yin Wang, Sujie Xu, Shijun Ma, Kexin Yang, Zhijun Liu, and Weidong Wang. Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. *Computer networks*, 190:107969, 2021.

[Wang *et al.*, 2023] Runqing Wang, Gang Wang, Jian Sun, Fang Deng, and Jie Chen. Flexible job shop scheduling via dual attention network-based reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):3091–3102, 2023.

[Wang *et al.*, 2025] Zunxun Wang, Junqing Li, Xiaolong Chen, Peiyong Duan, and Jiake Li. Uncertain interruptibility multiobjective flexible job shop via deep reinforcement learning based on heterogeneous graph self-attention. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[Wu *et al.*, 2025] Lincong Wu, Xiaoxia Li, Xin Lu, Zaiwen Feng, and Yanguo Jing. An adaptive meta-reinforcement learning framework for dynamic flexible job shop scheduling. *IEEE Transactions on Automation Science and Engineering*, 22:24036–24052, 2025.

[Xie *et al.*, 2019] Jin Xie, Liang Gao, Kunkun Peng, Xinyu Li, and Haoran Li. Review on flexible job shop scheduling. *IET collaborative intelligent manufacturing*, 1(3):67–77, 2019.

[Xu *et al.*, 2021] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. Industry 4.0 and industry 5.0—inception, conception and perception. *Journal of manufacturing systems*, 61:530–535, 2021.

[Xu *et al.*, 2024] Yuanxing Xu, Mengjian Zhang, Ming Yang, and Deguang Wang. Hybrid quantum particle swarm optimization and variable neighborhood search for flexible job-shop scheduling problem. *Journal of Manufacturing Systems*, 73:334–348, 2024.

[Yao *et al.*, 2024] Youjie Yao, Qihao Liu, Ling Fu, Xinyu Li, Yanbin Yu, Liang Gao, and Wei Zhou. A novel mathematical model for the flexible job-shop scheduling problem with limited automated guided vehicles. *IEEE Transactions on Automation Science and Engineering*, 2024.

[Yuan *et al.*, 2024] Erdong Yuan, Liejun Wang, Shuli Cheng, Shiji Song, Wei Fan, and Yongming Li. Solving flexible job shop scheduling problems via deep reinforcement learning. *Expert Systems with Applications*, 245:123019, 2024.

[Zhang *et al.*, 2024] Wenquan Zhang, Fei Zhao, Yong Li, Chao Du, Xiaobing Feng, and Xuesong Mei. A novel collaborative agent reinforcement learning framework based on an attention mechanism and disjunctive graph embedding for flexible job shop scheduling problem. *Journal of Manufacturing Systems*, 74:329–345, 2024.

[Zhao *et al.*, 2023] Linlin Zhao, Jiaxin Fan, Chunjiang Zhang, Weiming Shen, and Jing Zhuang. A drl-based reactive scheduling policy for flexible job shops with random job arrivals. *IEEE Transactions on Automation Science and Engineering*, 21(3):2912–2923, 2023.

[Zhao *et al.*, 2025] Peng Zhao, You Zhou, Di Wang, Zhiguang Cao, Yubin Xiao, Xuan Wu, Yuanshu Li, Hongjia Liu, Wei Du, Yuan Jiang, et al. Dual operation aggregation graph neural networks for solving flexible job-shop scheduling problem with reinforcement learning. In *Proceedings of the ACM on Web Conference 2025*, pages 4089–4100, 2025.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.