Graph-RHO: Critical-path-aware Heterogeneous Graph Network for Long-Horizon Flexible Job-Shop Scheduling

Yujie Li
Beijing University of Posts and Telecommunications
Beijing, China
liyujie2003@bupt.edu.cn

Jiuniu Wang
City University of Hong Kong
Hong Kong, China
wangjiuniu@gmail.com

Mugen Peng
Beijing University of Posts and Telecommunications
Beijing, China
pmg@bupt.edu.cn

Guangzuo Li
Chinese Academy of Sciences
Beijing, China
ligz@aircas.ac.cn

Wenjia Xu
Beijing University of Posts and Telecommunications
Beijing, China
xuwenjia@bupt.edu.cn

***Abstract*—Long-horizon Flexible Job-Shop Scheduling (FJSP) presents a formidable combinatorial challenge due to complex, interdependent decisions spanning extended time horizons. While learning-based Rolling Horizon Optimization (RHO) has emerged as a promising paradigm to accelerate solving by identifying and fixing invariant operations, its effectiveness is hindered by the structural complexity of FJSP. Existing methods often fail to capture intricate graph-structured dependencies and ignore the asymmetric costs of prediction errors, in which misclassifying critical-path operations is significantly more detrimental than misclassifying non-critical ones. Furthermore, dynamic shifts in predictive confidence during the rolling process make static pruning thresholds inadequate. To address these limitations, we propose Graph-RHO, a novel critical-path-aware graph-based RHO framework. First, we introduce a topology-aware heterogeneous graph network that encodes subproblems as operation-machine graphs with multi-relational edges, leveraging edge-feature-aware message passing to predict operation stability. Second, we incorporate a critical-path-aware mechanism that injects inductive biases during training to distinguish highly sensitive bottleneck operations from robust ones. Third, we devise an adaptive thresholding strategy that dynamically calibrates decision boundaries based on online uncertainty estimation to align model predictions with the solver's search space. Extensive experiments on standard benchmarks demonstrate that Graph-RHO establishes a new state of the art in solution quality and computational efficiency. Remarkably, it exhibits exceptional zero-shot generalization, reducing solve time by over 30% on large-scale instances (2000 operations) while achieving superior solution quality. Our code is available at https://github.com/IntelliSensing/Graph-RHO.**

***Index Terms*—Graph neural network, flexible job-shop scheduling, multi-task learning.**

I. Introduction

The Flexible Job-Shop Scheduling Problem (FJSP) is a cornerstone of modern smart manufacturing and logistics, governing the efficiency of resource allocation in complex production systems [1]. In FJSP, jobs composed of sequentially constrained operations must be scheduled onto a candidate machine pool, requiring simultaneous decisions on machine assignment and execution order for every operation. In industrial practice, these problems manifest as long-horizon tasks, necessitating the optimization of thousands of operations with intricate precedence constraints over extended time horizons. While exact solvers and meta-heuristics [2], [3] can handle small-scale instances, they suffer from the "curse of dimensionality" in long-horizon settings, where the combinatorial search space expands exponentially, rendering global optimization computationally intractable.

To tackle this scalability challenge, Rolling Horizon Opti-

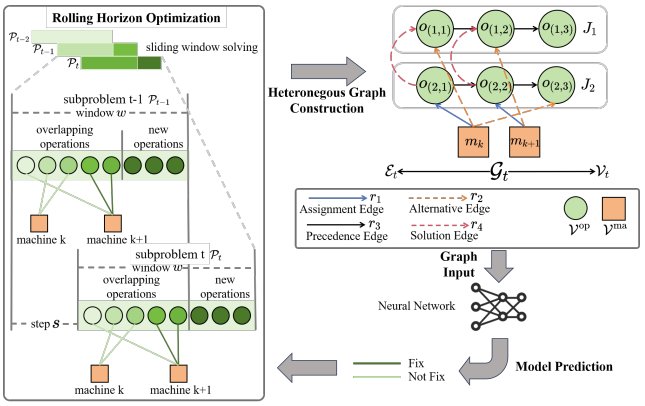

Fig. 1. **Illustration of Graph-RHO.** Within the RHO loop, each subproblem is mapped to a heterogeneous graph that explicitly encodes topological constraints via multi-relational edges. Leveraging this structural representation, the neural network identifies stable operations (solid dark green lines) and fixes their machine assignments.

This work has been funded by the National Natural Science Foundation of China under Grant 62301063.

mization (RHO) [4]–[6] has emerged as a standard paradigm. As shown in Fig. 1, to reduce global complexity, RHO decomposes the problem into manageable subproblems and solves them iteratively using a sliding window that advances in fixed steps, to cover both overlapping and new operations. However, a critical bottleneck arises because conventional RHO repeatedly re-optimizes all overlapping operations, even though a substantial portion of them preserve stable prior operation-machine assignments, leading to significant computational redundancy. To mitigate this, the recent Learning-based RHO framework L-RHO [7] pioneered a data-driven approach that uses an MLP network to identify and fix stable operation assignments, thereby successfully pruning the search space. However, limited by its vector-based representation, the MLP network abstracts interactions among operations and machines into simplified linear dependencies, which inadequately capture the underlying physical constraints and structural properties of FJSP.

Building upon this paradigm, we draw inspiration from the intrinsic graph-structured nature of FJSP, where operation sequences and machine constraints are deeply intertwined, as shown in Fig. 1. We observe that the stability of overlapping operations is fundamentally governed by these complex topological dependencies rather than simple statistical correlations. This insight motivates the construction of Graph Neural Networks to explicitly model the job shop floor as a relational graph. By interpreting the structural relationships between precedence and resource contention, we can achieve significantly more precise identification and fixation of invariant operations within the rolling windows.

Motivated by these, we propose Graph-RHO, a novel, critical-path-aware **Graph**-based **RHO** framework that advances the learning-based decomposition paradigm. Specifically, we introduce a topology-aware heterogeneous graph encoder to align the representation mechanism with the intrinsic graph-structured nature of FJSP. By explicitly modeling the shop floor via edge-feature-aware message passing, this module captures the constraint patterns embedded in the topology, enabling the model to reason about complex dependencies rather than merely fitting statistical correlations. Moreover, we introduce a critical-path-aware mechanism that incorporates a critical path identification objective during training. This auxiliary task injects an inductive bias, forcing the model to distinguish highly sensitive bottleneck operations from robust ones, thereby ensuring robust search-space pruning. Furthermore, we devise an adaptive thresholding inference strategy that dynamically aligns the pruning threshold with the model's online uncertainty distribution, creating a deep synergy between the neural predictor and the combinatorial solver.

Our main contributions are summarized as follows:

- We propose Graph-RHO, a learning-based RHO framework that leverages a heterogeneous graph encoder to explicitly model the FJSP shop floor. Through edge-feature-aware message passing, this module captures multi-relational topological constraints, significantly enhancing the representation capability for complex scheduling dynamics.
- We introduce a critical-path-aware mechanism. By incorporating an auxiliary critical path identification training objective, we inject an inductive bias that promotes the model to distinguish high-sensitivity critical operations from robust ones, thereby safeguarding solution quality against erroneous pruning.
- We devise an adaptive thresholding inference strategy. By dynamically calibrating decision boundaries based on online uncertainty estimation, this mechanism establishes a self-adjusting synergy between the neural predictor and the combinatorial solver, effectively balancing pruning efficiency with feasibility.
- Extensive benchmarking against a comprehensive suite of exact, meta-heuristic, and learning-based baselines confirms that Graph-RHO establishes a new state-of-the-art across various long-horizon FJSP settings. Crucially, the model demonstrates exceptional zero-shot generalization in both scale expansion and load intensification scenarios. Our model maintains superior solving efficiency and solution quality, effectively overcoming distribution shifts.

II. Related Works

A. Long Horizon FJSP

Traditional FJSP solvers, including exact methods [2] and meta-heuristics [3], struggle with the exponential time complexity of long-horizon scenarios. While Deep Reinforcement Learning (DRL) offers promise, constructive agents [8]–[10] often fail to scale effectively. To address this, Rolling Horizon Optimization (RHO) [11], [12] decomposes the problem into iterative overlapping subproblems. However, standard RHO suffers from computational redundancy by repeatedly re-optimizing invariant variables. The recent L-RHO framework [7] mitigates this by learning to fix stable operations within overlapping subproblem windows. Yet, L-RHO relies on topology-agnostic MLPs, failing to capture intrinsic graph constraints, which highlights the need for structurally aligned representations.

B. GNNs in Combinatorial Optimization

Graph Neural Networks (GNNs) have proven effective at capturing topological structures for Combinatorial Optimization. In scheduling, GNNs are widely used to encode disjunctive graphs for DRL-based dispatching [8], [13] or to enhance exact solvers via Learning-to-Branch [14], [15]. Despite these advances, the integration of GNNs into RHO remains unexplored. Existing learning-based RHO methods rely on MLPs, neglecting critical edge features like precedence and resource contention. Our work bridges this gap by introducing a heterogeneous graph network tailored for FJSP RHO, shifting the paradigm from statistical fitting to topological reasoning.

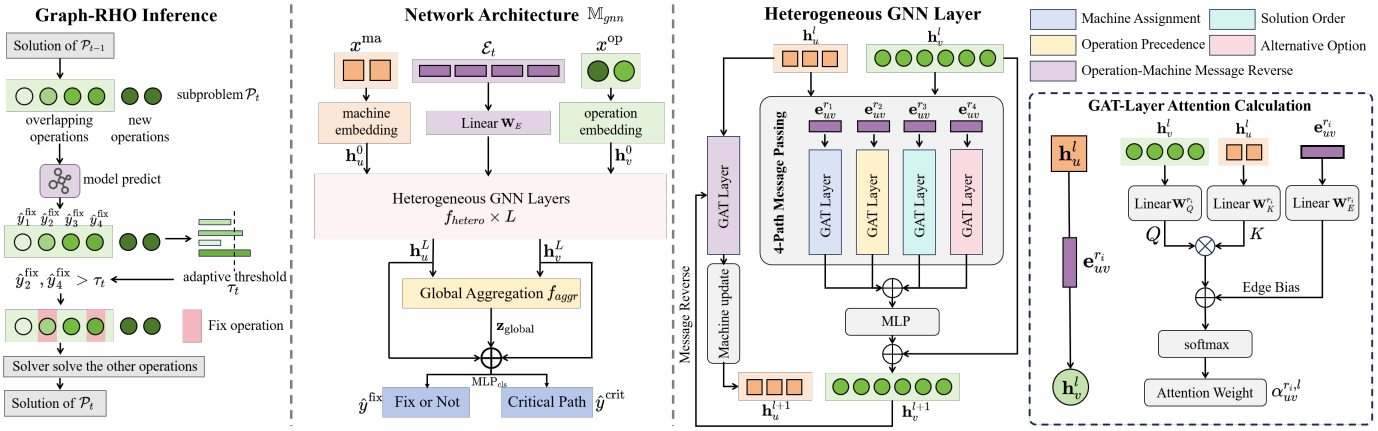

Fig. 2. **Overview of Graph-RHO: i) Graph-RHO Inference**: The model predicts stability probabilities for overlapping operations, employing an adaptive threshold τ_t to fix high-confidence variables and prune the search space for subproblem $\mathcal{P}_t$. **ii) Network Architecture**: The $\mathbb{M}_{gnn}$ processes heterogeneous graph states via stacked GNN layers and global aggregation (f_{aggr}) to simultaneously optimize operation stability ($\hat{y}^{\text{fix}}$) and path criticality ($\hat{y}^{\text{crit}}$) predictions. **iii) Heterogeneous GNN Layer**: The layer executes 4-path message passing to capture multi-relational dependencies, coupled with a reverse flow for machine updates. The **GAT Inset** illustrates the edge-feature-aware attention mechanism, where edge constraints $\mathbf{e}_{uv}^{r_i}$ are injected as structural bias.

III. Methodology

In this section, we present the design of our Graph-RHO, a learning-based RHO framework that accelerates long-horizon FJSP. We first formalize the Flexible Job-Shop Scheduling Problem and the Rolling Horizon Optimization paradigm in section III-A. Then detail the three pillars of our proposed method: the heterogeneous graph network $\mathbb{M}_{gnn}$ in Sec. III-C, the critical-path-aware mechanism $\mathbb{M}_{cpa}$ in Sec. III-D, and finally, the adaptive thresholding strategy $\mathbb{M}_{thr}$ for inference in Sec. III-E.

A. Problem Formulation

We model the Flexible Job-Shop Scheduling Problem (FJSP) as a discrete optimization problem defined by a tuple $\langle$machines, jobs, operations$\rangle$ as $\langle \mathcal{M}, \mathcal{J}, \mathcal{O} \rangle$. Here $\mathcal{M} = \{m_1, \ldots, m_{N_m}\}$ is a set of N_m machines and $\mathcal{J} = \{J_1, \ldots, J_{N_j}\}$ is a set of N_j jobs. Each job J_i is defined by an ordered sequence of operations $(o_{(i,1)}, \ldots, o_{(i,n_i)}) \in \mathcal{O}$, governed by the linear precedence constraints $o_{(i,k)} \rightarrow o_{(i,k+1)}$ for $k = 1, \ldots, n_i - 1$. For each operation $o_{(i,k)}$, the solver must make two interdependent decisions: **i)** Select a machine $m_{(i,k)} \in M_{(i,k)} \subseteq \mathcal{M}$ with deterministic processing time $p_{i,k}$; **ii)** Determine the start time $s_{i,k} \geq 0$ of each operation. The optimization goal is to find a valid schedule $\Pi = \{(m_{(i,k)}, s_{i,k}) \mid \forall o_{i,k} \in \mathcal{O}\}$ that minimizes the makespan $C_{\max} = \max_{i,k}(s_{i,k} + p_{i,k})$, which corresponds to the maximum completion time over all operations, subject to precedence constraints and machine capacity constraints.

To overcome the computational intractability of global optimization, Rolling Horizon Optimization (RHO) employs an iterative sliding-window mechanism. At each iteration t, RHO constructs a subproblem $\mathcal{P}_t$ containing w operations. Solving $\mathcal{P}_t$ yields a local schedule Π_t, from which only the immediate s operations are committed. The remaining $w - s$ operations form the overlap region, denoted as $\mathcal{O}_t^{overlap}$, which is traditionally deferred for full re-optimization. To mitigate the redundancy of repeatedly optimizing this region, the learning-based paradigm exploits the stability hypothesis [7], which posits that a subset of operations $\mathcal{O}_t^{fix} \subseteq \mathcal{O}_t^{overlap}$ retains invariant machine assignments across consecutive windows. By utilizing a model to predict and fix the variables in $\mathcal{O}_t^{fix}$ to their values from Π_{t-1}, the solver's search space for the next iteration is effectively pruned. To ensure precise identification of $\mathcal{O}_t^{fix}$, Graph-RHO employs a heterogeneous graph network to extract topological dependencies. Specifically, during inference (see in Fig. 2), the model predicts fixation scores for overlapping operations and applies an adaptive threshold τ_t to dynamically distinguish the most reliable subset, which are fixed to their assignments from Π_{t-1}, thereby pruning the search space for the solver.

B. Graph-GHO Framework

We propose the critical-path-aware heterogeneous **Graph**-based **R**olling **H**orizon **O**ptimization (Graph-RHO) framework. Graph-RHO synergizes three core advancements to identify stable operations: **i)** a topology-aware heterogeneous graph network $\mathbb{M}_{gnn}$ that explicitly encodes the intrinsic disjunctive graph structure via edge-feature-aware message passing; **ii)** a critical-path-aware mechanism $\mathbb{M}_{cpa}$ that incorporates critical path identification as an auxiliary task during training to rectify the sensitivity-agnostic limitation where standard classification treats all operations equally; and **iii)** a neural-symbolic synergy adaptive thresholding inference strategy $\mathbb{M}_{thr}$ that calibrates decision boundaries against predictive uncertainty shifts.

C. Heterogeneous Graph Network

Effective resolution of FJSP demands a precise understanding of the intricate topological associations between operation and machine nodes. To this end, we construct a heterogeneous graph network, denoted as $\mathbb{M}_{gnn}$ (Fig. 2). Structurally, this

model comprises stacked heterogeneous graph neural network layers f_{hetero} for constraint propagation, a global aggregation module f_{aggr} for system-level context integration, and a dedicated task head MLP_{cls}. In the following, we first detail the construction of the heterogeneous graph that encodes the subproblem state. Next, we introduce the stacked heterogeneous graph neural network layers f_{hetero}. Finally, we describe the global aggregation module f_{aggr} and task head for the ultimate prediction.

Heterogeneous Graph. At each iteration t, we map the state of the current subproblem $\mathcal{P}_t$ to a heterogeneous graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ indicating (nodes, edges), as illustrated in Fig. 1. In the following, we detail the construction of node features and heterogeneous edges to capture the system state.

The node set $\mathcal{V}_t = \mathcal{V}^{\text{op}} \cup \mathcal{V}^{\text{ma}}$ comprises two distinct types of entities, defined as follows: **i)** operation nodes $\mathcal{V}^{\text{op}}$: For each planed operation $o_{(i,k)}$, we construct a feature vector $x^{\text{op}} \in \mathbb{R}^{15}$ containing static attributes, such as processing times and job IDs, alongside dynamic states like the current start time $s_{i,k}$ and overlap status. **ii)** machine nodes $\mathcal{V}^{\text{ma}}$: For each machine $m_j \in \mathcal{M}$, we construct a feature vector $x^{\text{ma}} \in \mathbb{R}^{11}$ summarizing its workload statistics, including the average completion time and the number of assigned overlapping operations.

To decouple the complex constraints in FJSP, the edge set $\mathcal{E}_t$ consists of a set of relations $\mathcal{R}$ containing four distinct semantic edge types $\{r_1, r_2, r_3, r_4\}$. For an operation node v and its neighbor u, an edge $e_{uv}^{r_i} \in \mathcal{E}_t$ of relation type $r_i \in \mathcal{R}$ carries an edge feature vector $\mathbf{e}_{uv}^{r_i}$. These relations are defined as: **i)** machine assignment r_1 ($\mathcal{O} \rightarrow \mathcal{M}$): This represents the current tentative assignment of operation v to machine u. The edge feature includes the processing duration $p_{i,k}$ and an assignment indicator to encode direct resource occupation. **ii)** alternative option r_2 ($\mathcal{O} \rightarrow \mathcal{M}$): This links operation v to other compatible machines $u' \in \{M_{(i,k)} - \{u\}\}$, enabling the model to perceive the opportunity cost of the current assignment. **iii)** precedence constraint r_3 ($\mathcal{O} \rightarrow \mathcal{O}$): These are directed edges from $o_{(i,k-1)}$ to $o_{(i,k)}$ representing job-sequence constraints defined in $\mathcal{J}$. **iv)** solution order r_4 ($\mathcal{O} \rightarrow \mathcal{O}$): These directed edges represent the execution sequence on the same machine derived from the previous local schedule Π_{t-1}.

By unifying these node entities and semantic edges, $\mathbb{M}_{gnn}$ establishes a topology-complete representation that inherently possesses permutation invariance. This design empowers the model to effectively capture the intrinsic topological structure of FJSP. As corroborated by empirical results, this structural alignment endows the model with powerful topological perception capabilities and exceptional zero-shot generalization, enabling the learned policy to seamlessly transfer across varying scales without retraining.

Heterogeneous Graph Neural Network Layers. To effectively extract topological relation and simulate constraint propagation, $\mathbb{M}_{gnn}$ stacks L heterogeneous graph neural network layers f_{hetero}. Within each layer, we adopt a relation-specific edge-feature-aware message-passing mechanism in which operation nodes aggregate information from neighbors based on their distinct relational types $r_i \in \mathcal{R}$.

For a target operation, its scheduling status is intrinsically determined by the states of its topological neighbors, such as the current workload of compatible machines or the completion progress of preceding operations. To capture these interactions, we calculate attention coefficients to weigh the importance of each neighbor with graph attention networks (GATs) [16]. However, standard graph attention is insufficient for scheduling as it neglects the quantitative attributes of the connections. To accurately reflect physical constraints, we explicitly inject edge features $\mathbf{e}_{uv}^{r_i}$ into the attention computation [17]. This design acts as a structural bias, ensuring that neighbors with significant attributes, such as longer processing times, exert a stronger influence on the target node.

Formally, let $\mathbf{h}_v^l$ and $\mathbf{h}_u^l$ denote the embeddings of node v and neighbor u at layer l. For a target operation v, we compute the constraint-aware attention coefficient $\alpha_{uv}^{r_i,l}$ under relation r_i as follows (visualized in the **GAT-Layer Attention Calculation** inset of Fig. 2):

$$\text{score}_{uv}^{r_i,l} = \frac{(\mathbf{W}_Q^{r_i}\mathbf{h}_v^l)^\top(\mathbf{W}_K^{r_i}\mathbf{h}_u^l)}{\sqrt{d_k}} + \mathbf{W}_E^{r_i} \cdot \mathbf{e}_{uv}^{r_i}\,, \tag{1}$$

$$\alpha_{uv}^{r_i,l} = \text{softmax}_{u\in\mathcal{N}_{r_i}(v)}\left(\text{score}_{uv}^{r_i,l}\right)\,, \tag{2}$$

where $\mathbf{W}_Q^{r_i}$, $\mathbf{W}_K^{r_i}$ and $\mathbf{W}_E^{r_i}$ are learnable weight matrices, $\mathcal{N}_{r_i}(v)$ is the neighbors of the node v, and $\mathbf{W}_E^{r_i}$ projects the edge features into the attention space to modulate the connection weight dynamically.

With the attention coefficients established, operation nodes aggregate messages from all four relation types. These heterogeneous messages are concatenated and fused via a residual MLP to update the operation state $\mathbf{h}_v^{l+1}$:

$$\mathbf{m}_v^{r_i,l} = \sum_{u\in\mathcal{N}_{r_i}(v)} \alpha_{uv}^{r_i,l}\left(\mathbf{W}_V^{r_i}\mathbf{h}_u^l\right)\,, \tag{3}$$

$$\mathbf{h}_v^{l+1} = \text{Norm}\left(\mathbf{h}_v^l + \text{MLP}\left(\Big\|_{r_i\in\mathcal{R}} \mathbf{m}_v^{r_i,l}\right)\right)\,, \tag{4}$$

where $\|$ denotes concatenation.

Crucially, we introduce a reverse message passing step to complete the constraint loop. Machine nodes aggregate messages from their assigned operations via reversed assignment edges to update their own embeddings $\mathbf{h}_u^{l+1}$. This mechanism allows machines to dynamically reflect current congestion levels and broadcast this resource availability back to operations in the subsequent layer. By grounding predictions in this closed-loop structural causality, the model achieves robust reasoning about resource contention and precedence states within the local receptive field.

Prediction. After L message passing f_{hetero} layers, we obtain final embeddings for all nodes. Before the final prediction, we perform a global aggregation step in f_{aggr}. We apply average pooling across all operations and machine nodes to obtain a global context vector $\mathbf{z}_{\text{global}}$, which is then concatenated with the node-wise embeddings, ensuring that local decisions are conditioned on the global system load. For

each candidate operation $o_{(i,k)} \in \mathcal{O}_t^{overlap}$, we construct a final comprehensive representation $\mathbf{h}^{\text{final}}_{o_{(i,k)}}$ by concatenating its local operation embedding, the global context, and the embedding of its currently assigned machine $\mathbf{h}_u^L$:

$$\mathbf{h}^{\text{final}}_{o_{(i,k)}} = \left[\mathbf{h}_v^L; \mathbf{z}_{\text{global}}; \mathbf{h}_u^L\right] , \tag{5}$$

where $[;]$ means concatenation. Finally, the stability probability is predicted as $\hat{y}^{\text{fix}}_{i,k} = \sigma(\text{MLP}_{\text{cls}}(\mathbf{h}^{\text{final}}_{o_{(i,k)}}))$, where σ is the activation function. By leveraging the explicit topological graph, this probability reflects a rigorous assessment of the operation's stability within the complex constraint network rather than a mere heuristic estimation based on local statistics.

D. Critical-Path-Aware Mechanism

While the heterogeneous graph network $\mathbb{M}_{gnn}$ captures topological structures, training solely on binary stability classification treats all operations homogeneously, ignoring the asymmetric cost of decision errors in FJSP. According to the Critical Path Method (CPM) principles [18], misclassifying a critical operation (where total slack is 0) directly degrades the makespan, whereas errors on non-critical nodes are often absorbed by temporal flexibility. To address this, we introduce a critical-path-aware mechanism, $\mathbb{M}_{cpa}$, that adds an auxiliary critical-path-aware task during training. This mechanism injects a bottleneck-oriented inductive bias into the latent space, which explicitly guides the model to prioritise representations that distinguish high-sensitivity bottlenecks from robust operations.

We define operation criticality based on total slack. For an operation $o_{(i,k)}$, the earliest start times $S^E_{i,k}$ and latest start times $S^L_{i,k}$ derive from the disjunctive graph of the current local schedule Π_t. The slack $S_{i,k}$ is defined as $S_{i,k} = S^L_{i,k} - S^E_{i,k}$. An operation is labeled critical if $S_{i,k} = 0$, indicating that it lies on the longest path for which any delay strictly increases the makespan.

Supervision Data Collection. To support the training, we generate supervision signals from two distinct views: **i)** operation fix labels y^{fix}: We strictly adhere to the self-labeling protocol defined in L-RHO [7], where an operation is labeled stable $y^{\text{fix}}_{i,k} = 1$ if its machine assignment remains consistent with a lookahead oracle schedule. **ii)** criticality labels y^{crit}: In contrast, we derive auxiliary labels by analyzing the topological properties of the default schedule Π_t^{default}. By executing forward and backward passes on the disjunctive graph to compute the total slack, we assign $y^{\text{crit}}_{i,k} = 1$ if $S_{i,k} < \epsilon$ indicating the operation lies on the longest path, and $y^{\text{crit}}_{i,k} = 0$ otherwise.

Dual-Head Architecture. As is shown in Fig. 2, to inject the training objective, we employ a dual-head architecture sharing the structure-aware embeddings. The main head MLP_{fix} predicts the stability probability $\hat{y}^{\text{fix}}_{i,k}$, while the auxiliary head MLP_{crit} predicts the criticality probability $\hat{y}^{\text{crit}}_{i,k}$.

Training Objective. The training objective combines the operation fix loss $\mathcal{L}_{\text{fix}}$ and the criticality loss $\mathcal{L}_{\text{crit}}$. Both are formulated as binary cross-entropy losses over the overlap operations $\mathcal{O}_t^{overlap}$:

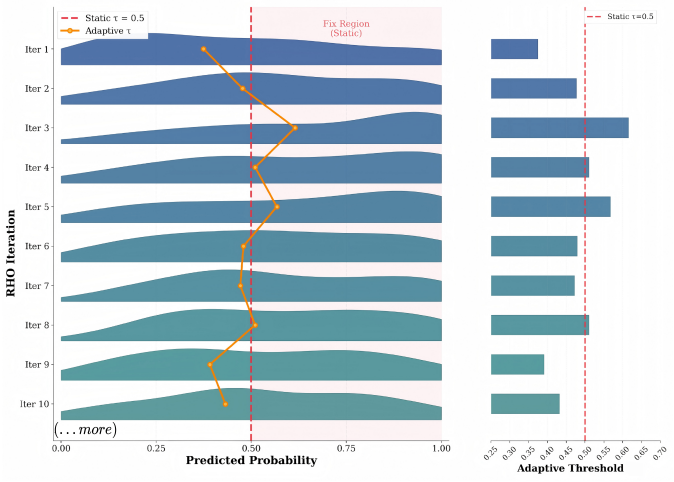

Fig. 3. **Temporal Distribution Shift and Thresholding Strategies. (Left)** The Ridgeline plot illustrates the evolving density of predicted fix probabilities across RHO iterations. A static threshold (red line) fails to adapt, whereas our adaptive threshold (orange line) dynamically tracks the distribution's tail. **(Right)** The resulting adaptive threshold values fluctuate to maintain a consistent fixing ratio.

$$\mathcal{L}_{\text{fix}} = -\frac{1}{|\mathcal{O}_t^{overlap}|} \sum_{o_{(i,k)}} \left[y^{\text{fix}}_{i,k} \log \hat{y}^{\text{fix}}_{i,k} + (1 - y^{\text{fix}}_{i,k}) \log(1 - \hat{y}^{\text{fix}}_{i,k})\right] \tag{6}$$

$$\mathcal{L}_{\text{crit}} = -\frac{1}{|\mathcal{O}_t^{overlap}|} \sum_{o_{(i,k)}} \left[y^{\text{crit}}_{i,k} \log \hat{y}^{\text{crit}}_{i,k} + (1 - y^{\text{crit}}_{i,k}) \log(1 - \hat{y}^{\text{crit}}_{i,k})\right] \tag{7}$$

Here, $\hat{y}_{i,k}$ is the predicted probability and $y_{i,k}$ is the label. The final joint loss is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{fix}} + \lambda \mathcal{L}_{\text{crit}} \tag{8}$$

where λ balances the tasks. Optimizing $\mathcal{L}_{\text{crit}}$ acts as a regularizer, penalizing the encoder for discarding topological information about the critical path, thereby ensuring prudent decisions at high-sensitivity nodes.

E. Adaptive Thresholding Strategy

Relying on a static probability threshold (e.g., $\tau = 0.5$) overlooks the dynamic uncertainty inherent in the rolling horizon process. As visualized in Fig. 3, the probability density of stability predictions exhibits significant temporal drift across iterations. A static threshold imposes a rigid cutoff on these evolving distributions: when the distribution shifts left, it results in insufficient pruning that burdens the solver; when it shifts right, excessive fixing degrades solution quality.

To bridge this gap, we propose the adaptive thresholding strategy $\mathbb{M}_{thr}$ that treats the model's output as a relative ranking and fixes the top subset of operations determined by a pre-set ratio γ. For each iteration t, we first sort the predicted fix probabilities of N overlapping operations $\mathcal{O}_t^{overlap}$ in descending order, denoted as $p_{(1)} \geq p_{(2)} \geq \cdots \geq p_{(N)}$.

TABLE I

FJSP UNDER MAKESPAN OBJECTIVE. EACH COLUMN CORRESPONDS TO A DIFFERENT FJSP SIZE GIVEN BY THE FORMAT: TOTAL NUMBER OF OPERATIONS, FOLLOWED BY A TUPLE OF $(N_m, N_j, N_{ops/job})$, I.E., NUMBER OF MACHINES $|\mathcal{M}|$, JOBS $|\mathcal{J}|$, AND OPERATIONS PER JOB. "TRANSFER" MEANS TRANSFER FROM THE 1200 CASE WITHOUT FINE-TUNING.

	600 (10, 20, 30)		**800** (10, 20, 40)		**1200** (10, 20, 60)		**2000** (10, 20, 100), Transfer	
	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓
CP-SAT (10 hours) [2]	36000	1583 ± 65	36000	2128 ± 75	36000	3206 ± 87	36000	18821 ± 1986
CP-SAT (30 minutes) [2]	1800	2274 ± 147	1800	4017 ± 413	1800	10925 ± 1013	1800	39585 ± 2707
GA [3]	1800	3659 ± 87	1800	5150 ± 92	1800	8086 ± 111	1800	10406 ± 640
ARD-LNS (Time-based) [19]	300	2100 ± 269	400	3298 ± 365	600	5115 ± 558	1000	14258 ± 1522
ARD-LNS (Machine-based) [19]	300	3974 ± 633	400	6594 ± 1227	600	12764 ± 4206	1000	52225 ± 1793
Oracle-LNS (Time-based) [20]	300	1789 ± 104	400	2663 ± 120	600	4470 ± 151	1000	7275 ± 785
DRL-Echeverria, Greedy [21]	84 ± 27	2347 ± 189	115 ± 32	3105 ± 161	105 ± 7	4570 ± 218	213 ± 10	7673 ± 356
DRL-Echeverria, Sampling [21]	157 ± 8	2351 ± 139	217 ± 11	3131 ± 165	427 ± 20	4657 ± 192	603 ± 41	7774 ± 271
DRL-Ho, Greedy [22]	4 ± 2	3030 ± 69	6 ± 4	4038 ± 87	11 ± 6	6045 ± 99	22 ± 3	10044 ± 115
DRL-Ho, Sampling [22]	174 ± 1	2907 ± 46	168 ± 14	3907 ± 56	385 ± 12	5870 ± 71	517 ± 10	9848 ± 92
DRL-20K, Greedy [23]	4 ± 0.02	1628 ± 72	6 ± 0.04	2128 ± 80	10 ± 0.1	3141 ± 97	16 ± 0.2	5184 ± 114
DRL-20K, Sample 100 [23]	29 ± 0.3	1551 ± 60	47 ± 1	2048 ± 69	110 ± 3	3063 ± 81	302 ± 10	5082 ± 98
DRL-20K, Sample 500 [23]	146 ± 5	1537 ± 61	261 ± 8	2031 ± 68	597 ± 16	3045 ± 79	1738 ± 16	5062 ± 98
Default RHO (Long) [4]	599 ± 55	1529 ± 58	728 ± 98	2044 ± 75	1099 ± 108	3002 ± 87	2871 ± 244	4994 ± 114
Default RHO [4]	244 ± 21	1558 ± 73	348 ± 26	2103 ± 78	545 ± 36	3136 ± 91	862 ± 42	5207 ± 114
Warm Start RHO [4]	203 ± 23	1521 ± 67	278 ± 22	2055 ± 75	420 ± 33	3081 ± 96	716 ± 41	5057 ± 106
L-RHO [7]	126 ± 19	1513 ± 70	160 ± 23	2015 ± 86	259 ± 37	3011 ± 106	473 ± 52	4982 ± 132
Graph-RHO (Ours)	**98 ± 21**	**1493 ± 73**	**139 ± 29**	**1985 ± 80**	**189 ± 29**	**2972 ± 97**	**321 ± 30**	**4938 ± 119**

To maintain a target fixation ratio γ, we define the adaptive threshold τ_t as the k-th largest probability value:

$$\tau_t = p_{(k)}, \quad \text{where } k = \lfloor \gamma \cdot N \rfloor \tag{9}$$

Here, $\lfloor \cdot \rfloor$ denotes the floor function, and $p_{(k)}$ serves as the dynamic decision boundary. The set of fixed operations is then explicitly given by the top-k candidates: $\mathcal{O}_t^{fix} = \{o_i \mid p_i \geq \tau_t\}$. To ensure robustness against uniformly low-confidence states, we further apply a safety floor τ^{min} as $\tau_t^{safe} = \max(\tau_t, \tau^{min})$.

Ultimately, this rank-based approach effectively decouples pruning decisions from fluctuations in absolute confidence. By consistently fixing the most reliable top-ranked subset defined by the ratio, Graph-RHO guarantees a stable reduction of the combinatorial search space, thereby maintaining a robust equilibrium between solving efficiency and solution feasibility throughout the rolling process.

IV. EXPERIMENT

In this section, we conduct comprehensive empirical evaluations to validate the effectiveness and generalization capability of our Graph-RHO framework. Our experiments are designed to answer the following core research questions:

- **Main Performance (Sec. IV-B).** Does Graph-RHO outperform baselines in solution quality and efficiency across various long-horizon FJSP settings?
- **Zero-Shot Generalization (Sec. IV-C).** Can the model generalize to larger scales (**Scale Generalization**) and higher loads (**Load Robustness**) without fine-tuning?
- **Ablation Studies (Sec. IV-D).** What are the contributions of the heterogeneous graph network $\mathbb{M}_{gnn}$, critical-path-aware mechanism $\mathbb{M}_{cpa}$, and adaptive thresholding strategy $\mathbb{M}_{thr}$?

A. Implementation

The Heterogeneous GNN encoder is configured with a hidden dimension of $d = 64$ and a depth of $L = 2$ layers, with 4 attention heads per GAT. We employ sigmoid activation functions and apply a dropout rate of 0.1 to mitigate overfitting. The model is trained using the AdamW optimizer with a batch size of 64 for 200 epochs. The learning rate is initialized at 1×10^{-4} and decayed using a cosine annealing scheduler. To balance the two training objectives, the weight assigned to the auxiliary critical-path-aware task is set to $\lambda = 0.5$.

For the RHO setup, we fix the planning window size at $w = 80$ and the execution step size at $s = 30$. During inference, the confidence-adaptive thresholding strategy employs a target fixation ratio of $\gamma = 0.6$, subject to a safety floor of $\tau^{\min} = 0.3$. The subproblems are solved using the OR-Tools CP-SAT solver [2], configured with a linearization level of 2 to enhance constraint propagation efficiency.

B. Main Results on FJSP with Makespan Objective

Experimental Setup and Baselines. We adhere to the protocol established in L-RHO [7], benchmarking on FJSP (makespan) using distributions from DANIEL [23] extended to significantly larger horizons. We evaluate three standard problem sizes $(N_m, N_j, N_{\text{ops/job}})$ with configurations of $(10, 20, 30/40/60)$, totaling 600–1,200 operations. Following this protocol, L-RHO and our Graph-RHO are both trained on a dataset of 450 instances and evaluated on 100 test instances. Additionally, we evaluate zero-shot transferability on a large-scale setting $(10, 20, 100)$, comprising 2,000 operations, using the model trained on 1,200-operation scale.

Graph-RHO is compared against four baseline categories: 1) **Global Solvers:** The exact solver CP-SAT (30 min/10 hr) and the meta-heuristic GA [3]; 2) **Decomposition Heuris-**

TABLE II

COMPARISON OF GENERALIZATION CAPABILITIES ACROSS UNSEEN PROBLEM SCALES AND LOAD DENSITIES. THE MODEL IS TRAINED SOLELY ON THE SMALL-SCALE $(10, 20, 30)$ SETTING AND DIRECTLY EVALUATED ON LARGER OR DENSER INSTANCES WITHOUT FINE-TUNING.

	Scale Generalization				**Load Robustness**			
	$(15, 30, 30)$		$(20, 40, 30)$		$(10, 30, 30)$		$(10, 40, 30)$	
Method	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓
Default RHO	179 ± 19	1455 ± 78	166 ± 9	1539 ± 84	354 ± 25	2176 ± 86	471 ± 30	2864 ± 85
L-RHO	72 ± 6	1451 ± 69	76 ± 7	1529 ± 78	196 ± 27	2162 ± 88	213 ± 36	2860 ± 105
Graph-RHO (Ours)	**42 ± 4**	**1435 ± 67**	**46 ± 3**	**1453 ± 75**	**142 ± 19**	**2144 ± 89**	**160 ± 20**	**2831 ± 101**

tics: ARD-LNS (Time/Machine-based) [19] and Oracle-LNS (Time-based); 3) **Constructive DRL:** State-of-the-art solvers including DRL-Echeverria [21], DRL-Ho [22], and DRL-20K [23] (Greedy/Sampling); and 4) **RHO Methods:** Default, Warm Start, and the previous SOTA L-RHO [7].

Results and Analysis. TABLE I summarizes the performance. Graph-RHO establishes a superior frontier between solution quality and computational efficiency. While constructive DRL baselines (e.g., DRL-Ho and DRL-20K in Greedy mode) achieve the lowest latency, they suffer from severe quality degradation, yielding makespans that are significantly worse than RHO-based methods (e.g., DRL-Ho's makespan is nearly double that of Graph-RHO on 2,000-operation scale). In contrast, Graph-RHO delivers state-of-the-art solution quality that rivals or surpasses heavy iterative solvers (like CP-SAT 10h) on large instances, while remaining orders of magnitude faster. This validates the fundamental advantage of the rolling horizon decomposition for long-horizon scheduling.

Crucially, our Graph-RHO significantly outperforms Default RHO and the previous SOTA L-RHO in both efficiency and quality. Graph-RHO achieves substantial speedups, particularly in the zero-shot transfer setting $(10, 20, 100)$. It reduces solve time by 32.1% compared to L-RHO (473s $\rightarrow$ 321s). This efficiency gain is attributed to the heterogeneous GNN encoder, which captures topological constraints more effectively than the MLP used in L-RHO. The structure-aware embeddings enable more precise stability predictions on unseen large-scale graphs, allowing the solver to prune the search space more aggressively without risking feasibility. Moreover, Graph-RHO consistently yields lower makespans than L-RHO (e.g., 1493 vs. 1513 on 600-operation scale). This quality improvement primarily stems from the critical-path-aware auxiliary task. Unlike standard binary classification objectives, which treat all operations indiscriminately, our model explicitly learns to identify and protect critical operations from being erroneously fixed, ensuring that local pruning decisions do not compromise the global makespan.

C. Zero-shot Generation Test Results

Experimental Setup. We evaluate zero-shot transferability by applying models trained exclusively on small-scale $(10, 20, 30)$ instances directly to unseen scenarios without fine-tuning. We define two distinct protocols: **i) Scale Generalization:** We expand the problem dimensions to $(15, 30, 30)$ and $(20, 40, 30)$ while maintaining a constant job-to-machine ratio ($N_j/N_m = 2$), thereby testing adaptability to graph expansion. **ii) Load Robustness:** We intensify resource contention by increasing job counts to 30 and 40 while fixing machine resources ($N_m = 10$), effectively elevating the load ratio to 3.0 and 4.0.

Results and Analysis. As is shown in TABLE II, Graph-RHO demonstrates exceptional zero-shot robustness, consistently outperforming both default RHO and L-RHO across all transfer settings. In the scale generalization test, we observe a clear "generalization collapse" in L-RHO. On the source distribution $(10, 20, 30)$ (see in TABLE I), L-RHO improves the makespan by 2.9% over default RHO. However, on the target $(20, 40, 30)$ instance, this advantage shrinks to a negligible 0.6% ($1539 \rightarrow 1529$), indicating its learned policy fails to scale. In contrast, Graph-RHO not only maintains but amplifies its advantage, achieving a 5.6% makespan improvement ($1539 \rightarrow 1453$) on the large-scale target. It also reduces inference time by 39.5% compared to L-RHO ($76s \rightarrow 46s$), proving that the heterogeneous GNN encoder captures scale-invariant topological rules rather than overfitting to specific problem sizes. In the load robustness test, Graph-RHO exhibits superior resilience to congestion. As the job-to-machine ratio doubles ($2.0 \rightarrow 4.0$) in the $(10, 40, 30)$ setting, the default solver struggles. Our Graph-RHO maintains high efficiency ($160s$), delivering a 66% speedup over default RHO ($471s$). More importantly, it widens the quality gap against L-RHO, reducing the makespan by 29 units ($2860 \rightarrow 2831$), whereas L-RHO barely outperforms the default baseline in this high-contention regime. The superior generalization stems from our GNN network's ability to learn scale-invariant topological rules, unlike L-RHO's reliance on statistical aggregations that succumb to distribution shifts. By capturing relative structural roles rather than absolute values, the learned policy seamlessly adapts to larger or denser graphs.

D. Ablation Studies

We perform a progressive ablation study to isolate the impact of our three core contributions: the heterogeneous GNN encoder $\mathbb{M}_{gnn}$, the critical-path-aware mechanism $\mathbb{M}_{cpa}$, and the adaptive thresholding strategy $\mathbb{M}_{thr}$. We incrementally integrate these components into the default RHO baseline to assess their additive benefits. As shown in Table III, we observe a clear cumulative performance gain across all problem

TABLE III

ABLATION STUDIES. WE INCREMENTALLY INTEGRATE THE THREE CORE CONTRIBUTIONS: HETEROGENEOUS GNN ENCODER, CRITICAL-PATH-AWARE MECHANISM, AND ADAPTIVE THRESHOLDING, TO EVALUATE THEIR INDIVIDUAL IMPACT. $\mathbb{M}_{gnn}$ INTRODUCES THE GNN ENCODER; $\mathbb{M}_{cpa}$ INTRODUCES THE CRITICAL-PATH-AWARE OBJECTIVE; $\mathbb{M}_{thr}$ APPLIES THE CONFIDENCE-AWARE INFERENCE STRATEGY.

	Core Contributions			**600**		**800**		**1200**		**2000 (Transfer)**	
Method	$\mathbb{M}_{gnn}$	$\mathbb{M}_{cpa}$	$\mathbb{M}_{thr}$	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓	Time (s) ↓	Makespan ↓
Default RHO (Baseline)	-	-	-	244 ± 21	1558 ± 73	348 ± 26	2103 ± 78	545 ± 36	3136 ± 91	862 ± 42	5207 ± 114
+ $\mathbb{M}_{gnn}$	✓	-	-	114 ± 19	1501 ± 72	143 ± 19	1995 ± 83	250 ± 36	2987 ± 99	415 ± 48	4962 ± 129
+ $\mathbb{M}_{gnn}$ + $\mathbb{M}_{cpa}$	✓	✓	-	114 ± 22	1498 ± 75	148 ± 21	1991 ± 76	232 ± 29	2982 ± 103	388 ± 36	4951 ± 124
Graph-RHO (Full)	✓	✓	✓	$\mathbf{99 \pm 21}$	$\mathbf{1493 \pm 73}$	$\mathbf{139 \pm 29}$	$\mathbf{1985 \pm 80}$	$\mathbf{189 \pm 29}$	$\mathbf{2972 \pm 97}$	$\mathbf{321 \pm 30}$	$\mathbf{4938 \pm 119}$

scales. The introduction of the $\mathbb{M}_{gnn}$ (Row 2) consistently accelerates inference and improves the solution quality compared to the baseline (e.g., 59% time reductions and 5% makespan reductions on 800-operation scale), confirming that structural encoding is fundamental for topological feature extraction. Adding the critical-path-aware mechanism $\mathbb{M}_{cpa}$ (Row 3) acts as a robust stabilizer, yielding consistent makespan reductions (e.g., 6.5% time reduction and 0.2% makespan reduction on 2,000-operation scale) by preventing the model from erroneously fixing bottleneck operations. Moreover, integrating adaptive thresholding $\mathbb{M}_{thr}$ into the full Graph-RHO model delivers the largest efficiency boost, slashing solve time by over 17% on the largest 2,000-operation transfer task compared to the static thresholding variant, without compromising solution quality. This validates that the synergy of topological reasoning, critical-path-aware mechanism, and adaptive thresholding inference is essential for pushing the frontier of long-horizon FJSP scheduling.

V. CONCLUSION

In this work, we propose Graph-RHO, a critical-path-aware graph-based RHO framework for long-horizon FJSP. By synergizing a topology-aware heterogeneous graph encoder with edge-feature-aware message passing, a critical-path-aware auxiliary training objective, and an adaptive thresholding inference strategy, Graph-RHO establishes a new state of the art in both solution quality and computational efficiency across various FJSP settings, while exhibiting strong zero-shot generalization on large unseen instances without fine-tuning.

REFERENCES

[1] L. Thames and D. Schaefer, "Software-defined cloud manufacturing for industry 4.0," *Procedia cirp*, vol. 52, pp. 12–17, 2016.

[2] Google Developers, "CP-SAT Solver — OR-Tools," https://developers.google.com/optimization/cp/cp_solver/, 2024, accessed: 2025-11-26.

[3] X. Li and L. Gao, "An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem," *International Journal of Production Economics*, vol. 174, pp. 93–110, 2016.

[4] L. Glomb, F. Liers, and F. Rösel, "A rolling-horizon approach for multi-period optimization," *European Journal of Operational Research*, vol. 300, no. 1, pp. 189–206, 2022.

[5] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 52–65, 2011.

[6] S. Sethi and G. Sorger, "A theory of rolling horizon decision making," *Annals of operations research*, vol. 29, no. 1, pp. 387–415, 1991.

[7] S. Li, W. Ouyang, Y. Ma, and C. Wu, "Learning-guided rolling horizon optimization for long-horizon flexible job-shop scheduling," *arXiv preprint arXiv:2502.15791*, 2025.

[8] W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible job-shop scheduling via graph neural network and deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1600–1610, 2022.

[9] M. Zhang, L. Wang, F. Qiu, and X. Liu, "Dynamic scheduling for flexible job shop with insufficient transportation resources via graph neural network and deep reinforcement learning," *Computers & Industrial Engineering*, vol. 186, p. 109718, 2023.

[10] D. Huang, H. Zhao, W. Tian, and K. Chen, "A deep reinforcement learning method based on a multiexpert graph neural network for flexible job shop scheduling," *Computers & Industrial Engineering*, vol. 200, p. 110768, 2025.

[11] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[12] G. Lu, J. Ning, X. Liu, and Y. M. Nie, "Train platforming and rescheduling with flexible interlocking mechanisms: An aggregate approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 159, p. 102622, 2022.

[13] F. Teichteil-Königsbuch, G. Povéda, G. G. de Garibay Barba, T. Luchterhand, and S. Thiébaux, "Fast and robust resource-constrained scheduling with graph neural networks," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 33, 2023, pp. 623–633.

[14] R. Wang, Z. Zhou, K. Li, T. Zhang, L. Wang, X. Xu, and X. Liao, "Learning to branch in combinatorial optimization with graph pointer networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 1, pp. 157–169, 2024.

[15] A. G. Labassi, D. Chételat, and A. Lodi, "Learning to compare nodes in branch and bound with graph neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 000–32 010, 2022.

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] J. E. Kelley Jr and M. R. Walker, "Critical-path planning and scheduling," in *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, 1959, pp. 160–173.

[19] D. Pacino and P. Van Hentenryck, "Large neighborhood search and adaptive randomized decompositions for flexible jobshop scheduling," in *Proceedings of the International Joint Conference on Artificial Intelligence*. AAAI Press, 2011.

[20] T. Huang, J. Li, S. Koenig, and B. Dilkina, "Anytime multi-agent path finding via machine learning-guided large neighborhood search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 9, 2022, pp. 9368–9376.

[21] I. Echeverria, M. Murua, and R. Santana, "Solving large flexible job shop scheduling instances by generating a diverse set of scheduling policies with deep reinforcement learning," *arXiv preprint arXiv:2310.15706*, 2023.

[22] K.-H. Ho, J.-Y. Cheng, J.-H. Wu, F. Chiang, Y.-C. Chen, Y.-Y. Wu, and I.-C. Wu, "Residual scheduling: A new reinforcement learning approach to solving job shop scheduling problem," *IEEE Access*, vol. 12, pp. 14 703–14 718, 2024.

[23] R. Wang, G. Wang, J. Sun, F. Deng, and J. Chen, "Flexible job shop scheduling via dual attention network-based reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 3091–3102, 2023.