A scalable reinforcement learning framework for task allocation in flexible construction scenarios

Yifan Wang[1], Bin Yang[1]

[1] College of Civil Engineering, Tongji University, China
2410444@tongji.edu.cn, yangbin@tongji.edu.cn

Abstract

The construction process requires close collaboration among multiple crews, which relies fundamentally on efficient task allocation. However, the performance and efficiency of existing scheduling approaches remain limited and are insufficient to support fine-grained decision-making. This paper proposes a scalable Reinforcement Learning (RL) framework for Multi-Crew Task Allocation (MCTA) in flexible construction scenarios. First, a disjunctive graph is developed to model construction tasks and their internal constraints, upon which the MCTA problem is formulated as a Markov Decision Process. Then, a four-module RL network architecture is designed to address the MCTA problem, and the overall policy is trained using Proximal Policy Optimization (PPO). Case study shows that the proposed framework can effectively allocate tasks, achieving a 8.63% reduction in project duration and a 15.88% decrease in crew idle time compared with existing methods.

Keywords –

Construction scheduling; Task allocation; Reinforcement learning; Disjunctive graph; Proximal policy optimization

1 Introduction

The construction process involves multiple work crews that can independently execute specific tasks. The completion of these tasks collectively leads to the final delivery of the entire project. The organization of the construction workflow plays a crucial role in ensuring project efficiency[1,2]. Construction Schedule Optimization (CSO) aims to enhance this organization by systematically planning and coordinating the timing and sequence of construction activities. In this process, the overall project is decomposed into several manageable and analyzable task units, commonly referred to as the Work Breakdown Structure (WBS). Based on this, the sequence of task execution is determined under the constraints of process precedence and resource availability, and each task is assigned to corresponding work crews for execution.

The problem of determining the optimal start times together with executor assignment under resource and precedence constraints is defined as the Flexible Resource-Constrained Project Scheduling Problem (FRCPSP)[3,4], which is known to be NP-hard[5]. In existing studies, such problems are primarily addressed using mathematical methods[6,7], heuristic methods[8,9] and metaheuristic approaches[10,11]. With the increasing complexity and expanding scale of modern construction projects, the construction process has evolved into a dynamically changing complex system characterized by intricate spatiotemporal constraints[9,12] and expansion of project scale. However, these methods rely on static rules or complex computational mechanisms, resulting in insufficient performance and generalization in flexible construction scenarios with strong dynamics of tasks and resources[13,14].

Reinforcement Learning (RL) is an effective approach for solving sequential decision-making problems in dynamic environments. Several studies have applied RL to determine construction project schedules[15–17]. However, these studies exhibit two key limitations: (1) they suffer from limited scalability, which leads to the curse of dimensionality in large-scale problems and an inability to handle variable problem sizes caused by dynamic changes or unexpected events; (2) they often fail to explicitly model the internal spatiotemporal constraints among construction tasks, forcing the neural network to implicitly learn project-specific decision rules in a black-box manner. These limitations hinder the practical applicability of RL in complex and flexible construction scenarios and restrict its ability to generate high-quality, fine-grained schedules.

To address these limitations, this study proposes an RL framework for modeling and solving fine-grained construction scheduling problem Multi-Crew Task Allocation (MCTA) problem from the perspective of Multi-Agent Simulation (MAS)[18]. The main contributions of this study are summarized as follows: (1)

A disjunctive graph is developed to model MCTA by explicitly representing spatiotemporal constraints among tasks. (2) A scalable RL framework is proposed to learn scheduling policies over the disjunctive graph, integrating Relational Graph Convolutional Network (RGCN) [19] and self-attention-based aggregation.

2 Related Work

2.1 Mathematical methods for construction scheduling

Early studies addressed the CSO problem using mathematical methods, among which the most common and influential is the Critical Path Method (CPM)[20]. CPM is a project scheduling technique that predicts the total project duration and identifies critical activities by locating the longest path of dependent tasks within a project. However, CPM has proven insufficient to handle the multi-objective, multi-constraint nature of modern scheduling problems, particularly in balancing resource continuity and cost trade-offs.

Consequently, Integer Programming (IP) and Linear Programming (LP) have been introduced to solve CSO problems. For example, Mattila et al.[21] proposed an integer linear programming-based linear scheduling model for resource leveling. Gomar et al.[22] developed a linear programming model to optimize the allocation of multi-skilled labor resources, minimizing labor costs while satisfying project demands. Ipsilands et al.[23] formulated a multi-objective linear programming model for scheduling linear and repetitive projects. To address the dynamic nature of the construction process, Robinson[24] applied a Dynamic Programming (DP) approach to determine project schedules that minimize overall duration, enabling the optimal allocation among activities with arbitrary cost－time functions.

2.2 Heuristic and metaheuristic methods for construction scheduling

Large-scale CSO problems are typically NP-hard, making it difficult for exact algorithms to obtain optimal solutions within a reasonable computation time. Heuristic methods, on the other hand, can efficiently produce near-optimal and satisfactory solutions by designing problem-specific decision rules. For example, Elazouni[25] proposed a heuristic algorithm that enumerates and ranks parallel project schedules under credit constraints, aiming to minimize project duration while maintaining real-time cash flow matching. Gajpal and Elazouni[26] introduced an enhanced heuristic algorithm based on a polynomial-shifting strategy, which replaces exhaustive search and reduces computation time from exponential to polynomial order without compromising solution accuracy. Similarly, Al-Shihabi et al.[27] developed a heuristic method that balances project duration and financial progress through a dual-priority rule enabling rapid generation of feasible scheduling solutions.

Metaheuristic methods improve candidate solutions through iterative computation based on predefined criteria, without requiring strong assumptions about the underlying CSO problem. Chan et al.[28] proposed a Genetic Algorithm (GA)-based construction resource scheduling method, which learns the structural features of project networks through selection and recombination mechanisms. Elazouni et al.[29] employed GA, Simulated Annealing (SA), and the Shuffled Frog-Leaping Algorithm (SFLA) to solve project scheduling problems under financial constraints, achieving the shortest total project duration with SA in large-scale networks. Ye et al.[11] further developed a two-stage task allocation framework for multiple heterogeneous construction robots based on an improved genetic algorithm, effectively enhancing collaboration efficiency in multi-robot systems.

2.3 RL for construction scheduling

The reliance of heuristic and metaheuristic approaches on predefined rules and static optimization limits their adaptability to the dynamic construction processes. RL is an effective approach for solving dynamic decision-making problems[30], and has been applied to address CSO challenges. Soman et al.[16] proposed a hybrid method that integrates RL with Linked-Data to automatically generate look-ahead schedules for construction projects. Kedir et al.[17] combined RL with agent-based modeling to obtain practical solutions that support decision-making in construction activity sequencing. Yuan et al.[15] considered both process precedence and resource constraints in bridge construction and employed RL to generate construction schedules. In addition, Agrawal et al.[31] developed a novel approach integrating Proximal Policy Optimization (PPO) with Monte Carlo sampling, which enables automatic generation of optimized assembly sequences and schedules for prefabricated buildings. These studies did not decompose construction projects into fine-grained task units and thus overlooked the precedence and spatial overlap constraints among tasks. Liu et al.[32] propose a Construction Markov Decision Process framework that leverages multi-agent RL to evaluate goals and plan paths, reducing project duration and labor idle time. However, they did not provide a systematic formulation of the scheduling problem, and therefore the methods still exhibit limitations in both performance and practical applicability.

3 Methodology

3.1 Problem formulation

This study focuses on a flexible construction process involving multiple crews, each possessing distinct skill sets, and models each crew as an agent. The entire project is decomposed into multiple tasks using WBS[33], and all agents collaboratively complete the construction through interleaved task execution, following the workflow illustrated in Figure 1. In this research, the MCTA problem incorporates three types of constraints, which are designed to explicitly capture the interactions among the three key elements in MCTA: tasks, labor resources, and workspace. Specifically, capability constraints reflect the matching relationship between tasks and crew skills, precedence constraints model the logical dependencies among tasks, and spatial overlap constraints characterize the conflicts arising from limited workspace.

(1) Capability constraint: Each crew can only execute tasks corresponding to its specific trade (e.g., a rebar-tying crew cannot perform formwork installation tasks).

(2) Precedence constraint: A task cannot be started until all its adjacent predecessor tasks are completed (e.g., formwork installation must precede concrete pouring).

(3) Spatial overlap constraint: If the workspace of a target task overlaps with that of a neighboring ongoing task, the target task cannot be initiated.

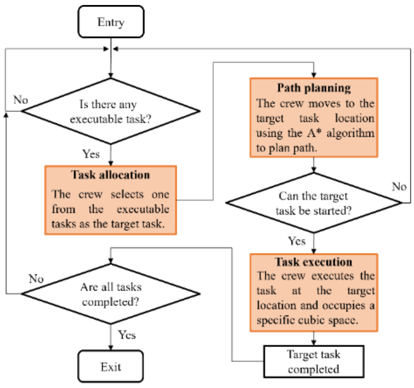


Figure 1 Workflow of crew agents

3.2 RL mathematical Model for MCTA

To address the MCTA problem, we adopt a RL approach, which requires formulating the scheduling process as a Markov Decision Process (MDP). The MDP can be formally described by a tuple $\langle \mathbf{S}, \mathbf{A}, \mathrm{P}, \mathrm{R}, \gamma \rangle$. $\mathbf{S}$ and $\mathbf{A}$ denote the state and action spaces, respectively, while $\mathbf{P}$ is the state transition probability function. R represents the reward function used to evaluate task allocation performance, and γ is the discount factor.

3.2.1 State

The state of the MCTA is defined by tasks attributes, agent attributes, and relationships among tasks. Attributes of tasks and agents are categorized into static and dynamic attributes, as illustrated in Table 1. $\mathbf{S}_{\mathrm{loc}}^{\mathrm{task}}$ and $\mathbf{S}_{\mathrm{loc}}^{\mathrm{agent}}$ are represented by two-dimensional coordinates on the plane. $\mathbf{S}_{\mathrm{type}}^{\mathrm{task}}$ and $\mathbf{S}_{\mathrm{type}}^{\mathrm{task}}$ are encoded as one-hot vectors, with the element corresponding to its type set to 1 and all others set to 0. $\mathbf{S}_{\mathrm{dur}} = [d]$ is determined based on the task quantity and the corresponding crew productivity rate. Four task pools— $task_{\mathrm{wait}}$, $task_{\mathrm{queue}}$, $task_{\mathrm{on}}$, and $task_{\mathrm{end}}$ —are used to represent the execution status of tasks[9]. $\mathbf{S}_{\mathrm{po}}$ is represented as a one-hot encoded vector, where the element corresponding to the current pool of a task is set to 1 and all others are set to 0.

Table 1 Summary of state space

Temporality	Attribute	Symbol	Representation
Static	Task location	$\mathbf{S}_{\mathrm{loc}}^{\mathrm{task}}$	2D vector
	Task type	$\mathbf{S}_{\mathrm{type}}^{\mathrm{task}}$	One-hot vector
	Execution time	$\mathbf{S}_{\mathrm{dur}}$	1D scalar
	Agent type	$\mathbf{S}_{\mathrm{type}}^{\mathrm{agent}}$	One-hot vector
Dynamic	Task pool	$\mathbf{S}_{\mathrm{po}}$	One-hot vector
	Agent location	$\mathbf{S}_{\mathrm{loc}}^{\mathrm{agent}}$	2D vector

The relationships between tasks can be represented with a disjunctive graph[34] $G = (T, P, S)$, as shown in Figure 2. Here, T denotes the set of all tasks (nodes in black). P is a set of directed edges representing the precedence constraints between consecutive tasks, directed from each predecessor task to its successor (illustrated in orange). S is a set of undirected edges, in which each edge connects a pair of tasks that have a spatial overlap relationship (illustrated in blue).

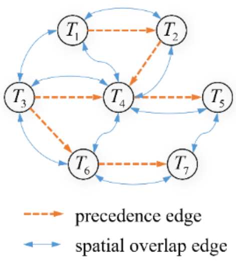


Figure 2 Disjunctive graph for MCTA

3.2.2 Action

Each crew agent outputs one discrete action a_t at a decision step t, where each optional action corresponds to a task. The dimensionality of the action space equals the total number of tasks. Due to the three types of constraints mentioned in Section 3.1, a large portion of the action space consists of invalid actions, which may lead to inefficient exploration during the agent's learning process. To address the problem, this study introduces action mask technique to handle a variable action space. Specifically, the logits of invalid actions are set to negative infinity at the output layer of the policy network, ensuring that their probabilities become zero after the softmax operation. Consequently, action sampling is performed only among valid actions, effectively improving exploration efficiency and training stability.

3.2.3 State transition

The state transition P of MCTA corresponds to the update of the overall construction progress. When an agent reaches an available task location, the task is moved from $task_{\text{queue}}$ to $task_{\text{on}}$. Upon task completion, the task transitions from $task_{\text{on}}$ to $task_{\text{end}}$, while its adjacent successor tasks are simultaneously moved from $task_{\text{wait}}$ to $task_{\text{queue}}$, thereby becoming eligible for execution.

3.2.4 Reward

The objective of MCTA is to minimize the total project duration, with all crew agents sharing a common reward defined as $\mathrm{R} = \mathrm{R}_{\text{finish}} + \mathrm{R}_{\text{time}}$. Here, $\mathrm{R}_{\text{finish}}$ is a terminal reward granted to all agents upon the completion of all construction tasks, while R_{time} is a stepwise penalty applied at each simulation timestep to encourage faster overall project completion.

3.3 Scalable RL network architecture for MCTA

This study proposes a scalable network architecture comprising four modules: a feature extraction module, a global representation module, a task scoring module, and a decoder model as illustrated in Figure 3.

3.3.1 Feature extraction module

The disjunctive graph represents precedence and spatial overlap constraints through different types of edges, thereby transforming spatiotemporal dependencies into connection features. To process such a heterogeneous graph with multiple edge types, a RGCN (parameterized by θ_{R}) is employed to extract task-level relational features.

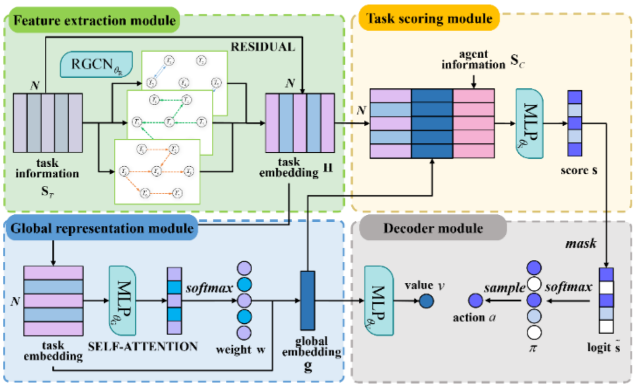


Figure 3 RL network architecture for MCTAFor a given task T_i, its raw features can be defined as:

$$\mathbf{S}_{T_i} = (\mathbf{S}_{\text{loc}}^{\text{task}} \parallel \mathbf{S}_{\text{type}}^{\text{task}} \parallel \mathbf{S}_{\text{dur}} \parallel \mathbf{S}_{\text{po}}) \tag{1}$$

where $\parallel$ is the vector concatenation operator. To mitigate over-smoothing problem during multi-layer information aggregation, the original features are concatenated with the RGCN output (Equation (2)) to form the d-dimensional embedding of n tasks $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n)^{\mathrm{T}} \in \mathbb{R}^{n \times d}$, completing the feature extraction process.

$$\mathbf{h}_i = (\mathbf{S}_{T_i} \parallel \mathrm{RGCN}_{\theta_{\mathrm{R}}}(\mathbf{S}_{T_i}; \theta_{\mathrm{R}})) \tag{2}$$

3.3.2 Global representation module

The amount of information increases linearly with the number of tasks, so task-level information must be compressed into a graph-level representation to prevent dimensionality explosion. This study proposes a self-attention–based global representation method to identify critical tasks and reduce information loss A multi-layer perceptron (MLP) parameterized by θ_{G} is used to compute the attention weight $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)^{\mathrm{T}}$ of each task based on its task embedding (Equation (3)(4)), after which a weighted average of all task embeddings is performed to obtain a global embedding $\mathbf{g}$ with the same dimensionality as each individual task embedding (Equation (5)).

$$w_i = \mathrm{MLP}_{\theta_{\mathrm{G}}}(\mathbf{h}_i; \theta_{\mathrm{G}}) \tag{3}$$

$$\alpha_i = \frac{\exp(w_i)}{\sum_{j=1}^{n} \exp(w_j)} \tag{4}$$

$$\mathbf{g} = \sum_{i=1}^{n} \alpha_i \mathbf{h}_i \in \mathbb{R}^d \tag{5}$$

3.3.3 Task scoring module

We designed a scalable task-shared scorer, which shares feature-mapping parameters across all tasks. For each task T_i, its task embedding $\mathbf{h}_i$ is concatenated with the global embedding $\mathbf{g}$ obtained from the global representation module and information $\mathbf{S}_c = (\mathbf{S}_{\text{type}}^{\text{agent}} \,\|\, \mathbf{S}_{\text{loc}}^{\text{agent}})$ of the agent, and then passed through an MLP scorer parameterized by θ_{S} to compute its score s_i. (Equation (6)).

$$s_i = \text{MLP}_{\theta_{\text{S}}}(\mathbf{h}_i \,\|\, \mathbf{g} \,\|\, \mathbf{S}_c; \theta_{\text{S}}) \tag{6}$$

Task scoring module not only preserves the independent comparability of individual nodes but also allows the network to learn the relative importance among tasks through backpropagation.

3.3.4 Decoder module

The decoder module utilizes the encoded information from upstream modules to generate both the value and action outputs. The value decoder maps the global information contained in the global embedding $\mathbf{g}$ into a scalar value v using MLP parameterized by θ_{V}:

$$v = \text{MLP}_{\theta_{\text{V}}}(\mathbf{g}; \theta_{\text{V}}) \in \mathbb{R} \tag{7}$$

The action decoder receives the task scores $\mathbf{s} = (s_1, s_2, ..., s_n)^{\text{T}}$ from the shared scorer. To handle the variable action space, invalid tasks are masked by setting their corresponding logits to negative infinity. Then softmax function is then applied to obtain the action probabilities. Finally, the selected action a_t is sampled from this probability distribution.

3.3.5 PPO for MCTA

PPO[35] is a widely used RL algorithm that improves training stability and efficiency by introducing a clipped objective function to limit abrupt policy updates. It alternates between sampling trajectories through agent–environment interaction and performing several epochs of gradient-based optimization, making it simple to implement yet highly effective for scheduling problems like MCTA. This study employs PPO to train agent strategy.

4 Case study

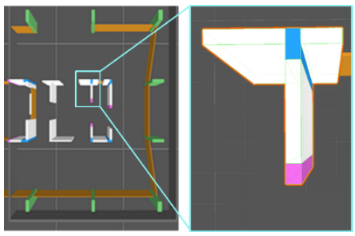
4.1 Case description and modeling process

To validate the effectiveness of the proposed method, we adopts a case study based on vertical component construction for a prefabricated structure. The case involves 7 construction crews (N=7) across 5 trade types, and a total of 86 construction tasks (n=86) associated with 36 structure components(Table 2). The 7 crews consist of one Joint cutting and Cleaning Crew (JCC), one Hoisting and Installation Crew (HIC), one Grouting Crew(GC), two Reinforcing Crews (RC1 and RC2), and two Form-work Crews (FC1 and FC2). Under capability constraints, each crew can only perform tasks corresponding to its trade. The simulation parameters are configured in strict accordance with reference[32]. We developed an RL environment based on Unity 3D.

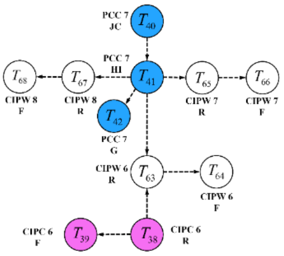
Table 2 Basic attributes of components

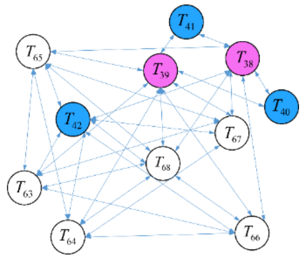
Component type	ID	Required tasks
Prefabricated columns	PCC 1-6	JC,HI,G
Cast-in-place columns	CIPC 1-5	R,F
Prefabricated walls	PCW 1-8	JC,HI,G
Cast-in-place walls	CIPW 1-17	R,F

A local structure segment (shown in Figure 4(a)) is selected to demonstrate the disjunctive graph modeling process. Within this segment, 11 tasks are involved, connected by 10 precedence relations, which are directed from predecessor tasks to successor tasks to represent precedence constraints, as shown in Figure 4(b). These precedence relationships are derived not only from general construction knowledge but also from the connection relationships among components and the specific technical requirements of the project. The 34 spatial overlap edges are illustrated in Figure 4(c). Each edge indicates that the corresponding tasks have overlapping bounding volumes and therefore cannot be executed simultaneously under the spatial overlap constraint. In Figure 4(b) and (c), the nodes are arranged using different layouts to better visualize the distinct connection patterns of the two relations.



(a) Selected segment of the structure





(b) Precedence relations (c) Spatial overlap relations

Figure 4 Modeling the case with disjunctive graph

4.2 Configurations

The simulation experiments were conducted on a computer equipped with a 13th Gen Intel® Core™ i9-13900K CPU (3.00 GHz), 64 GB RAM, an NVIDIA GeForce RTX 4090 GPU, and a 64-bit Windows 11 Professional operating system. In the simulation environment, each time step corresponds to 2.5 seconds, and each spatial unit represents one meter in the real-world project. Key network parameters are summarized in Table 3.

Table 3 Network parameters

Parameter	Value
Task feature dimension	12
Agent feature dimension	7
Number of RGCN layers	3
Hidden layer dimension of RGCN	64
Output layer dimension of RGCN	64
Embedding dimension	64+12=76
Number of MLP_{θ_G} layer	2
Hidden layer dimension of MLP_{θ_G}	64
Number of MLP_{θ_S} layer	3
Hidden layer dimension of MLP_{θ_S}	512
Number of MLP_{θ_V} layer	3
Hidden layer dimension of MLP_{θ_V}	512
Activation function	ReLU
Learning rate	0.0001
Epsilon	0.2
Number of epoch	3
Max step	330000
Buffer size	1280
Batch size	128

4.3 Results

Through iterative interaction with the environment (Figure 5), the proposed method successfully generated a fine-grained construction schedule for the case project. All 86 tasks were finished without constraints violated. Reference [36] evaluated 12 rules for task sequencing, from which we select three well-performing rules that are suited to MCTA as baselines. (1) First In First Out (FIFO): Tasks that enter the task pool $task_{\mathrm{queue}}$ earlier are given higher priority, reflecting a fairness-oriented scheduling strategy based on task availability. (2) Shortest Operation Time (SOT): Tasks with the shortest execution time are prioritized to improve efficiency. (3) Shortest Distance (SD): Tasks that are geographically closest to the crew agent are prioritized, aiming to reduce travel time. In addition, we import the Priority Rule (PR)[9], that comprehensively considers multiple factors.

Among metaheuristic methods, we selected GA as a baseline, as it is widely applied in scheduling research. Following the approach in [37], each gene represents the priority of a task, and the total project duration is used as the fitness function. The population size is set to 100. The optimization process is terminated after 80 generations, as the best fitness value shows no improvement for 20 consecutive generations. Also, an RL-based method in construction scheduling, FC-PPO[32], is included as a baseline. For fair comparison, this baseline adopts the same number and dimension of hidden layers as the scorer and value networks proposed in this study, and is trained for the same number of steps.

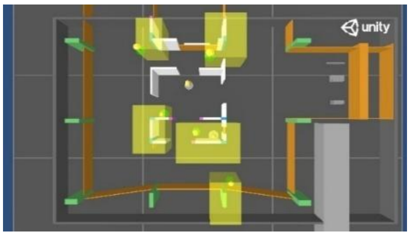


Figure 5 Runtime visualization of the simulator

Three performance indicators are defined to quantitatively evaluate the efficiency and stability of the MCTA system (as shown in Table 4), including:(1) Project Duration (PD): the total duration of the construction process.(2) Average Idle Time (AIT): the average idle time of crews caused by the unavailability of executable tasks.(3) Crew Utilization Rate (CUR): the ratio of working time to total time for crews. The results are averaged over 30 simulation runs with different random seeds.

Among the heuristic methods, FIFO, SOT, and SD each focus on a single factor and lack adaptability to complex construction scenarios. For highly dynamic construction scenarios, the GA method based on pre-calculated task priorities struggles to achieve reliable performance within limited iterations, with an average PD of up to 19.83 h. Among all seven evaluated methods, the proposed approach achieves the shortest average PD of 14.40 h, representing a 8.63 % reduction compared with the best competing method (FC-PPO). Moreover, its standard deviation is only 0.35 h, demonstrating excellent stability in performance. AIT and CUR reflect the utilization of human resources in the MCTA problem. The proposed method significantly reduces AIT of crews, achieving a 15.88 % reduction compared with the best among the other seven methods (FC-PPO), while also attaining the highest CUR. These results strongly demonstrate the superior performance of the proposed approach.

Table 4 Performance comparison of different methods

Method	PD (h)	AIT (h)	CUR (%)
FIFO	19.98±0.45	6.85±0.36	50.63±1.21
SOT	19.57±0.37	6.48±0.29	50.54±0.89
SD	16.30±1.04	4.88±0.61	56.68±2.15
PR	16.64±0.63	4.99±0.37	54.93±1.18
GA	19.83±0.27	6.79±0.23	50.75±0.65
FC-PPO	15.76±0.75	4.22±0.34	56.39±1.49
Ours	**14.40**±0.35	**3.55**±0.27	**60.29**±0.88

To further investigate whether the learned representations effectively capture real-world construction conditions, an additional analysis is conducted from two perspectives. (1) An ablation study is performed to evaluate the contribution of the graph-based representation. Specifically, the RGCN is replaced with a task-wise MLP that directly processes raw features. The results show that the cumulative reward does not exhibit an increasing trend during training, indicating that features extracted from the disjunctive graph are indispensable for decision-making. (2) A correlation analysis is conducted to examine the relationship between extracted features and the task importance reflected by the self-attention mechanism. The attention weights, together with the task scores, provide an indication of which features are most relevant to the construction conditions. The results show that the extracted features achieve an average Spearman rank correlation coefficient ρ of 0.46, with 70.0 % of feature dimensions exhibiting moderate or higher correlation (ρ > 0.4), and 17.5 % showing strong correlation (ρ > 0.7). In contrast, the raw features yield an average ρ of only 0.05, indicating a very limited contribution to representing construction conditions.

5 Conclusion and future work

This study proposes a disjunctive graph–based representation for flexible construction processes, together with a scalable RL framework to address the MCTA problem. The core network architecture integrates a feature extraction module, a task scoring module, a global representation module, and a decoder module. Within this architecture, a RGCN is employed to capture multi-relational features from the disjunctive graph, while a self-attention mechanism aggregates global information in a scale-agnostic manner. These designs enable the learned policy to generate high-quality task allocation decisions that shorten overall project duration and significantly improve labor resource utilization.

Future research should focus on developing large, standardized datasets for various construction types to learn generalized scheduling policies with stronger transferability and robustness. Advancing these directions will further strengthen the proposed framework's capability to support practical, real-time decision-making in complex construction management environments.

References

[1] FA. Agrama. Multi-objective genetic optimization for scheduling a multi-storey building. Automation in Construction, 44:119-128, 2014.

[2] M. Tomczak and P. Jaśkowski. Harmonizing construction processes in repetitive construction projects with multiple buildings. Automation in Construction, 139:104266, 2022.

[3] SH. Dike. Project scheduling with resource constraints. IEEE Transactions on Engineering Management, EM-11(4):155-157, 1964.

[4] H. Ding, C Zhuang, and J. Liu. Extensions of the resource-constrained project scheduling problem. Automation in Construction, 153:104958, 2023.

[5] M. Taghaddos, A. Mousaei, H. Taghaddos, U. Hermann, Y. Mohamed, and S. AbouRizk. Optimized variable resource allocation framework for scheduling of fast-track industrial construction projects. Automation in Construction, 158:105208, 2024.

[6] A. Kallantzis and S. Lambropoulos. Critical path determination by incorporating minimum and maximum time and distance constraints into linear scheduling. Engineering, Construction and Architectural Management, 11(3):211-222, 2004.

[7] AM. Elazouni and AA. Gab-Allah. Finance-based scheduling of construction projects using integer programming. Journal of Construction Engineering and Management, 130(1):15-24, 2004.

[8] W. Liu, J. Zhang, and W. Liu. Heuristic Methods for Finance-Based and Resource-Constrained Project Scheduling Problem. Journal of Construction Engineering and Management, 147(11):04021141, 2021.

[9] B. Liu, B. Yang, M. Dong, S. Jiang, and J. Xiao. Spatio-temporal deduction of floor construction based on the agent modeling of construction actors. Automation in Construction, 142:104487, 2022.

[10] N. Wongwai and S. Malaikrisanachalee. Augmented heuristic algorithm for multi-skilled resource scheduling. Automation in Construction, 20(4):429-445, 2011.

[11] X. Ye, H. Guo, and Z. Luo. Two-stage task allocation for multiple construction robots using an improved genetic algorithm. Automation in Construction, 165:105583, 2024.

[12] TQ. Nguyen, DKN. Chua. Preemptive Constraint Analysis in Construction Schedules. Journal of

Computing in Civil Engineering, 29(5):04014062, 2015.
[13] Q. Amarkhil and E. Elwakil. Enhanced planning and scheduling in building construction projects: an innovative approach to overcome scheduling challenges. International Journal of Construction Management, 24(16):1719-1729, 2024.
[14] L. Xie, Y. Chen, and R. Chang. Scheduling Optimization of Prefabricated Construction Projects by Genetic Algorithm. Applied Sciences, 11(12):5531, 2021.
[15] Y. Yao Y, VWY. Tam, J. Wang, KN. Le, A. Butera. Automated construction scheduling using deep reinforcement learning with valid action sampling. Automation in Construction, 166:105622, 2024.
[16] RK, Soman, M. Molina-Solana. Automating look-ahead schedule generation for construction using linked-data based constraint checking and reinforcement learning. Automation in Construction, 134:104069, 2022.
[17] NS. Kedir, S. Somi, AR. Fayek, and PHD. Nguyen. Hybridization of reinforcement learning and agent-based modeling to optimize construction planning and scheduling. Automation in Construction, 142:104498, 2022.
[18] A. Khodabandelu and J. Park. Agent-based modeling and simulation in construction. Automation in Construction, 131:103882, 2021.
[19] M. Schlichtkrull, TN. Kipf, P. Bloem, R. Berg, I. Titov, and M. Welling. Modeling Relational Data with Graph Convolutional Networks. arXiv, 1703.06103, 2017.
[20] JE. Kelley and MR. Walker. Critical-path planning and scheduling. In Eastern Joint IRE-AIEE-ACM Computer Conference, pages 160-173, Boston, USA, 1959.
[21] Mattila KG, Abraham DM. Resource leveling of linear schedules using integer linear programming. Journal of Construction Engineering and Management, 124(3):232-244, 1998.
[22] JE. Gomar, CT. Haas, and DP. Morton. Assignment and allocation optimization of partially multiskilled workforce. Journal of Construction Engineering and Management, 128(2):103-109, 2002.
[23] PG. Ipsilandis. Multiobjective linear programming model for scheduling linear repetitive projects. Journal of Construction Engineering and Management, 133(6):417-424, 2007.
[24] D. Robinson. Dynamic-Programming Solution to Cost-Time Tradeoff for Cpm. Management Science, 22(2):158-166, 1975.
[25] W. Liu, J. Zhang, W. Liu. Heuristic Methods for Finance-Based and Resource-Constrained Project Scheduling Problem. Journal of Construction Engineering and Management, 147(11):04021141, 2021.
[26] Y. Gajpal, A. Elazouni. Enhanced heuristic for finance-based scheduling of construction projects. Construction Management and Economics, 33(7):531-553, 2015.
[27] S. Al‐Shihabi, MM. AlDurgam. The contractor time‐cost‐credit trade‐off problem: integer programming model, heuristic solution, and business insights. International Transactions in Operational Research, 27(6):2841-2877, 2020.
[28] WT. Chan, DKH. Chua, and G. Kannan. Construction resource scheduling with genetic algorithms. Journal of Construction Engineering and Management, 122(2):125-132, 1996.
[29] A. Elazouni, A. Alghazi, SZ Selim. Finance-based scheduling using meta-heuristics: discrete versus continuous optimization problems. Journal of Financial Management of Property and Construction, 20(1):85-104, 2015.
[30] RS. Sutton and AG. Barto. Reinforcement Learning: An Introduction. 2nd ed. MIT Press, Cambridge, Massachusetts and London, England, 2018.
[31] AK. Agrawal, Y. Zou, H. Jin, M. Abdelmegid, and V. Gonzalez. Automated assembly sequence planning and scheduling in precast building projects using reinforcement learning. In Proceedings of the 42nd ISARC, Montreal, Canada, 2025.
[32] B. Yang, B. Liu, Y. Han Y, X. Meng, Y. Wang, H. Yang, and J. Xiao. Multi-agent reinforcement learning-enhanced decision-making of crew agents in floor construction processes. Advanced Engineering Informatics, 69:103996, 2026.
[33] B. Yang, B. Liu, J. Xiao, B. Zhang, Z. Wang, and M. Dong. A novel construction scheduling framework for a mixed construction process of precast components and cast-in-place parts in prefabricated buildings. Journal of Building Engineering, 43:103181, 2021.
[34] S. Wang, X. Li, L. Gao, J. Li, A multi-disjunctive-graph model-based memetic algorithm for the distributed job shop scheduling problem, Advanced Engineering Informatics 60 (2024)
[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. arXiv, 1707.06347, 2017.
[36] H.-H. Doh, J.-M. Yu, J.-S. Kim, D.-H. Lee, and S.-H. Nam. A priority scheduling approach for flexible job shops with multiple process plans. International Journal of Production Research, 51(12):3748–3764, 2013.
[37] M. Taghaddos, H. Taghaddos, U. Hermann, Y. Mohamed, and S. AbouRizk. Hybrid multi-mode simulation and optimization for subarea scheduling in heavy industrial construction. Automation in Construction, 125:103616, 2021.